AD-A118 613 COLORADO STATE UNIV FORT COLLINS F/G 12/1
VITERBI TRACKING OF RANDOMLY PHASE-MODULATED DATA (AND RELATED --ETC(U)
AUG 82 L L SCHARF DAAG29-79-C-0176
UNCLASSIFIED ARO-16437.9-EL NL

1 of 1

END
DATE
FILMED
9 82
DTIC

(11)

ARO 16437.9-EL

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER Final Report	2. GOVT ACCESSION NO. AD-A118613	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Viterbi Tracking of Randomly Phase-Modulated Data (and Related Topics)		5. TYPE OF REPORT & PERIOD COVERED Final : 9/1/79 to 2/15/82
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Louis L. Scharf		8. CONTRACT OR GRANT NUMBER(s) DAAG 29 79 C 0176
9. PERFORMING ORGANIZATION NAME AND ADDRESS Colorado State University Ft. Collins, CO 80521		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS U. S. Army Research Office Post Office Box 12211 Research Triangle Park, NC 27709		12. REPORT DATE August 10, 1982
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

NA

18. SUPPLEMENTARY NOTES

The view, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Phase; frequency; estimation; sequence estimation; maximum likelihood; dynamic programming; data communication; likelihood; autoregressive processes

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

In this final report we formulate the problems of phase and frequency estimation as fixed interval smoothing problems. Forward dynamic programming algorithms are derived to find maximum posterior sequence estimates that pass likely sequences through the data. The algorithms extend the threshold usually associated with phase lock loops. The algorithms are applied to simultaneous phase estimation and data decoding in phase jitter channels. The resulting error probabilities are the lowest currently achievable. → cont

82 08 26 023

cont

The results indicate that "there" are many problems in the domain of filtering and signal processing that can be profitably reformulated as sequence estimation problems.

Our interest in likelihood leads us from frequency tracking to exact likelihood for autoregressive moving average (ARMA) data. Fast Kalman filtering algorithms are derived for constructing exact likelihood in multivariable ARMA models. Several interesting connections are established between Wold, Kolmogorov, Wiener, and Kalman representations of stationary time series. Ideas are proposed for associating spectra with linear transformations.

Viterbi Tracking of Randomly Phase-Modulated Data

Final Report

Louis L. Scharf
University of Rhode Island
Kingston, RI 02881

August 5, 1982

Submitted to

U. S. ARMY RESEARCH OFFICE
P.O. Box 12211
Research Triangle Park, NC 27709

Contract/Grant Number

DAAG 29 79 C 0176

Colorado State University
Ft. Collins, Colorado 80523

Approved for Public Release;
Distribution unlimited.

The view, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

FOREWORD

Phase and frequency tracking problems comprise some of the most nettlesome nonlinear filtering problems in the realm of signal processing. These problems have held the interest of control and communication theorists at least since 1953/54 when Lehan and Parks, and Youla published their work on maximum likelihood and optimum demodulation on an interval. Over the years Cox, Viterbi, Cahn, Forney, and a host of others have advocated dynamic programming for the solution of nonlinear filtering problems. This research follows that tradition.

Dynamic Programming is advocated as a technique for finding the maximum a posteriori (MAP) phase or frequency modulated sequence to pass through a data set. The key idea is to pose a Markov chain model on the circle $(0,2\pi)$ for phase or frequency, and then generate candidate MAP sequences that are consistent with the data and the a priori probability structure.

TABLE OF CONTENTS

LIST OF APPENDIXES

INTRODUCTION

Phase and Frequency tracking are the classic nonlinear filtering problems. They arise in narrowband analog communication, data transmission, and spread spectrum communication. As usually stated, the problem is to obtain a causal estimate of the phase or frequency based on noisy phase modulated observations. The best known solutions are phase-locked loops (PLL's).

In any truly nonlinear filtering approach to optimum phase tracking, the basic problem is to propagate an a posteriori density, conditioned on an increasing measurement record, much as is done in Kalman Filtering. Unfortunately, there exist no finite-dimensional schemes for propagating the exact conditional density or for propagating a finite-dimensional sufficient statistic. One must approximate.

Under this contract we have developed an approach to phase and frequency sequence estimation (emphasis on the word sequence) that has its logical antecedents in the filtering philosophy of Youla and the data decoding philosophy of Viterbi. We have posed a maximum a posteriori probability (MAP) sequence estimation problem that leads to nonlinear MAP equations not unlike the continuous-time MAP interval equations. We have derived dynamic programming algorithms to efficiently solve for survivor phase and frequency sequences that approximate the desired MAP sequence. The algorithms also provide a handy mechanism for generating fixed-lag phase estimates, although this is not the problem for which the algorithm is derived.

In a loosely related set of problems we have studied exact likelihood for autoregressive moving average (ARMA) processes. We have derived fast algorithms for constructing likelihood, and established interesting connections between the work of Wold, Kolmogorov, Wiener, and Kalman. A fast Kalman filter has been realized in 16-bit arithmetic on an 8086 microprocessor.

In the sections to follow, we outline the problems studied and summarize important results.

STATEMENT OF PROBLEMS STUDIED

We summarize here the main problems studied under this contract.

Phase Modelling. For phase and frequency tracking the first problem to be studied is one of deriving suitable models for random phase and/or random frequency modulation.

Dynamic Programming Algorithm Development. Once a phase model is derived, the next problem is to derive a likelihood function and find a dynamic programming algorithm to find the maximum of likelihood.

Performance Evaluation for Phase and Frequency Sequence Estimation. The next problem is to simulate the dynamic programming algorithms on stochastic data and calculate Monte-Carlo performance results.

Simultaneous Phase Tracking and Data Decoding. When complex data are transmitted over phase jitter channels, there arises the problem of simultaneously tracking phase and decoding data symbols. The problem is to derive a joint likelihood function for phase and symbol sequences, maximize it with a dynamic program algorithm, and compute Monte-Carlo performance results.

Maximum Likelihood Identification of ARMA Systems. The problem here is to derive a fast algorithm to compute likelihood for autoregressive moving average (ARMA) sequences.

Fixed Point Implementation of Kalman Filters. The fast Kalman gain algorithm is a fixed point algorithm ideally suited for computation on a fixed point machine. But the problems of scaling and rounding remain. The question here is one of deriving scaling rules and calculating rounding error variances in time varying Kalman filters.

SUMMARY OF MOST IMPORTANT RESULTS

The most important results of this study are summarized below.

Phase Modelling. We have derived phase models for random phase, random FM, and random chirp modulation. Each model is a Markov chain defined on a cyclic group. Corresponding correlation and spectral results have been derived. The results generalize existing results on the spectral theory of chains, and leave one with the problem of selecting states, transition probabilities, and run lengths to achieve model matching with more conventional models. The results apply for coherent and noncoherent FM. The figure on the following page gives a geometric picture of the kinds of phase and frequency models we have used in most of our work on algorithm development, phase and frequency tracking, and simultaneous phase tracking and data decoding. See references 1,2, and 3 for additional details.

Dynamic Programming Algorithm Development. The basic measurement model in all of our work has been the following:

$$z_t = a_t e^{j\phi_t} + n_t$$

a_t : symbol drawn from a finite alphabet

ϕ_t : either a directly modulated phase sequence for which we know the transition probability density $p(\phi_{t+1}/\phi_t)$ or a function $\phi(\omega_t)$ of a frequency sequence ω_t for which we know the transition probabilty density $p(\omega_{t+1}/\omega_t)$

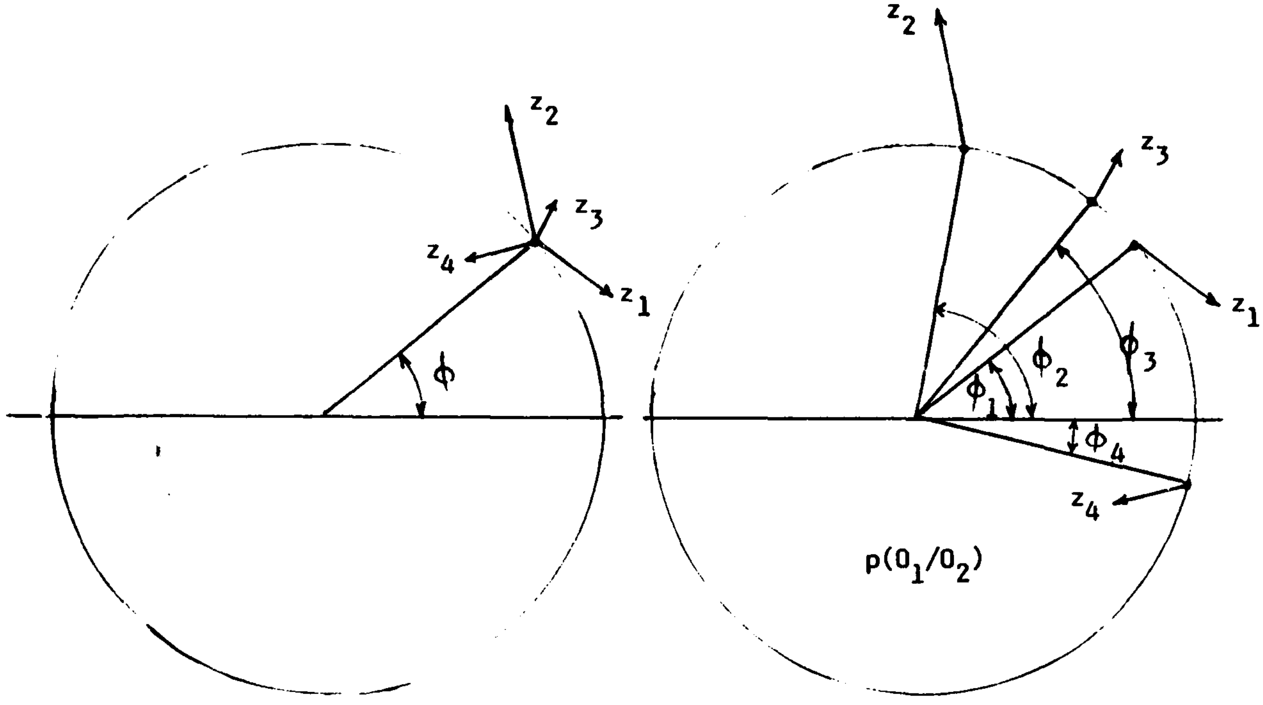
n_t : a sequence of independent and identically distributed normal random variables

With this model we have derived expressions for likelihood and found dynamic programming algorithms for exactly maximizing or approximately maximizing likelihood. Generally the algorithms take the form:

$$\max_{\phi_K,\phi_{K-1}} \; \max_{\phi_1,\ldots\phi_{K-2}} \left[L_{K-1} + \ln p(\phi_{K/K-1}) + g(\phi_K)\right]$$

The function g(.) depends on the details of the problem. See references 1,2, and 3 for details about selecting g(.) and implementing the algorithm on a finite trellis. The function L is likelihood.

It is our opinion that a variety of filtering problems in signal and image processing can be reformulated as sequence or interval estimation problems for which likelihood can be derived and for which algorithms can be found for approximating the maximum.

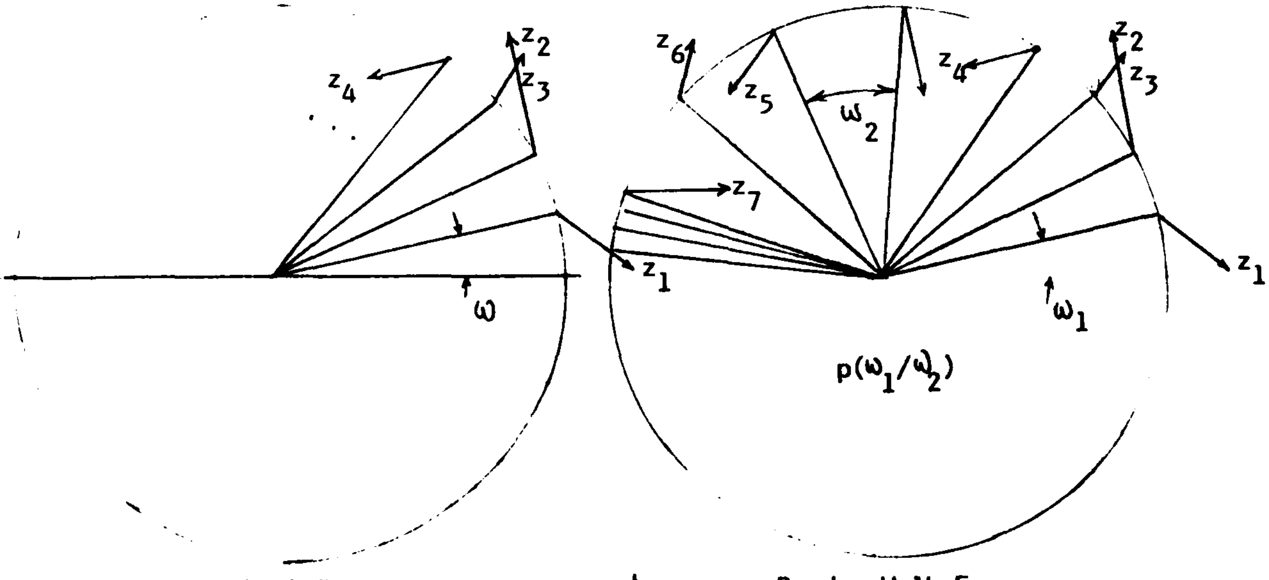


Constant Phase

Random Walk Phase

$$z_t = e^{j\phi_t} + n_t$$

$$Z_t \rightarrow (\hat{\phi}_1, \hat{\phi}_2, \ldots \hat{\phi}_K)$$



Constant Frequency

Random Walk Frequency

$$z_t = e^{j\phi_t} + n_t$$

$$\phi_t = \phi(\omega_t)$$

$$Z_t \rightarrow (\hat{\phi}_1, \hat{\phi}_2, \ldots \hat{\phi}_K)$$

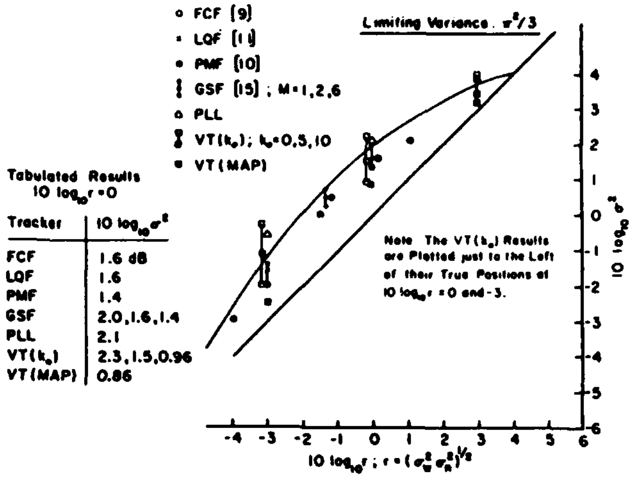
PHASE AND FREQUENCY MODELS

Performance Evaluation for Phase and Frequency Sequence Estimation. Our results are nicely summarized on the graphs of the following pages. The first compares estimation error variance for the dynamic programming (or Viterbi) solution with a host of other algorithms ranging from the phase lock loop to the point mass filter and the Fourier coefficient filter. The results apply to the problem of random walk phase tracking.

The next graph shows output SNR versus input CNR for sinusoidal modulation of a carrier. We have adapted our random walk FM frequency tracker to this problem and compared its performance with linear prediction trackers, and the trackers of Tufts and of Toomey and Short.

Our performance results indicate that sequence estimation by the method of dynamic programming to maximize likelihood on a finite trellis provides a way of improving on the performance of more classical causal estimators. This improvement can be significant at low SNR.

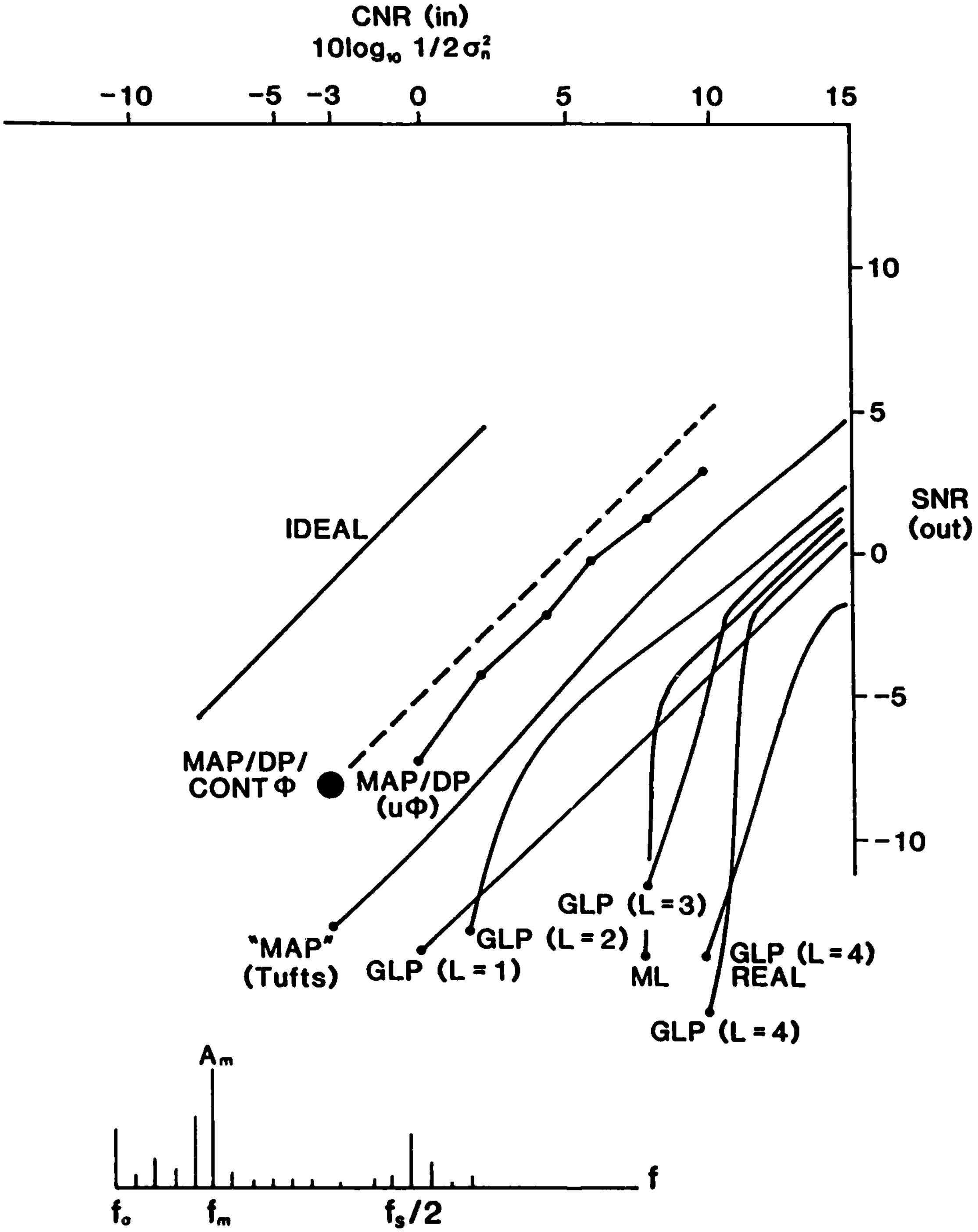
Simultaneous Phase Tracking and Data Decoding. The performance results for this problem are contained in Reference 2, where a variety of binary, phase shift keying, and quadrature shift keying communication problems are considered. The third figure in the sequence of three figures that appears on the next three pages shows just one of the many examples contained in Reference 2. The graph shows how two simultaneous phase trackers and data decoders, namely the Viterbi tracker and the jitter equalizer, achieve performance very close to that achievable under coherent phase conditions. The results apply to the decoding of 8-ary phase shift keyed symbols.



Tabulated Results
10 $\log_{10} r = 0$

Tracker	10 $\log_{10} \sigma^2$
FCF	1.6 dB
LQF	1.6
PMF	1.4
GSF	2.0, 1.6, 1.4
PLL	2.1
VT(k_o)	2.3, 1.5, 0.96
VT(MAP)	0.86

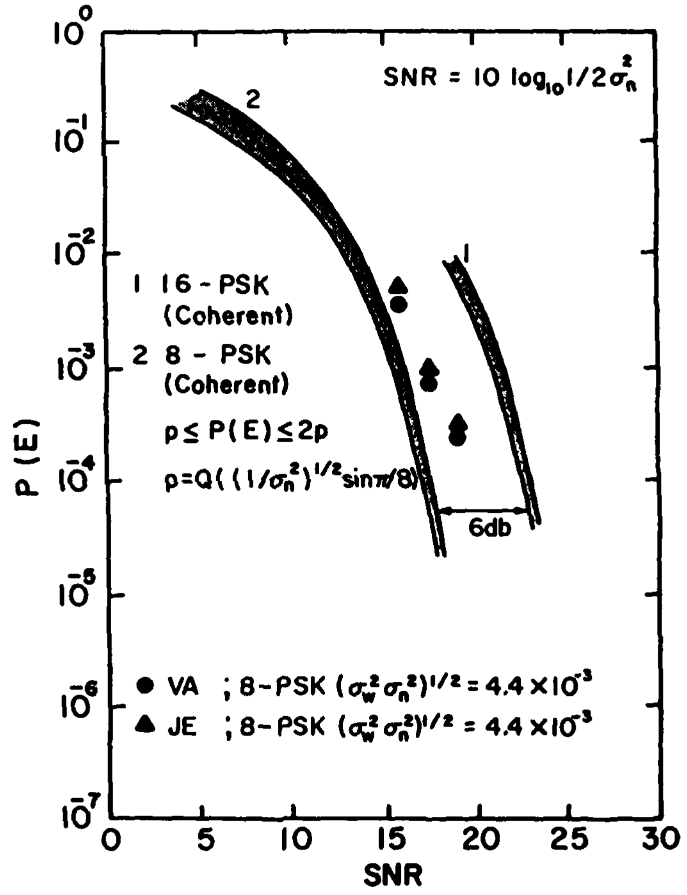
COMPARATIVE PERFORMANCE RESULTS FOR RANDOM PHASE TRACKING



DFT 105 pt sequence of estimates

$$SNR = 10\log_{10} \frac{A^2}{\sum_{1}^{n/2} A_i^2 - A_m^2}$$

COMPARATIVE PERFORMANCE RESULTS FOR SINUSOIDAL FM TRACKING



PERFORMANCE RESULTS FOR SIMULTANEOUS PHASE TRACKING AND DATA DECODING

Maximum Likelihood Identification of ARMA Systems. We have followed the lead of Akaike and Anderson and Moore to write down the innovations representation that reproduces the second order statistics of a stationary ARMA sequence. We have then associated the gain of a Kalman filter with the triangular square root of a Toeplitz matrix to rederive Morf's fast Kalman filter algorithm. The result is a fast algorithm for implementing likelihood. The results are summarized in References 4 and 5.

Fixed Point Implementation of Kalman Filters. Beginning with the innovations representation of a stationary ARMA sequence, we have derived scaling rules to prevent overflow in time varying Kalman filters and derived formulas for rounding error variance. The scaling rule is

$$\frac{q(k,k)^{\frac{1}{2}}}{s(k)} = \frac{\epsilon\, 2^{m-1}}{\delta}$$

$s(k)$: inverse of time varying scale constant

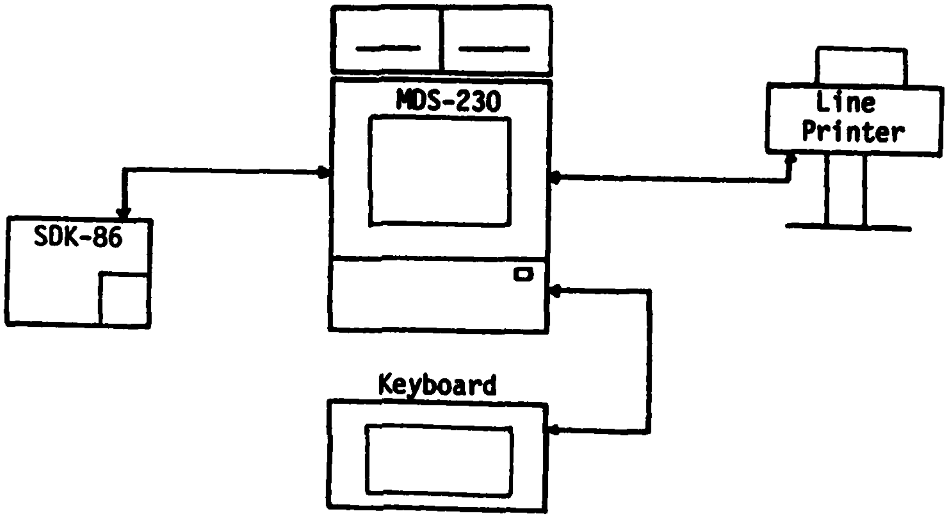
$q(k,k)^{\frac{1}{2}}$: (k,k)th element of the state variance matrix

$\epsilon\, 2^{m-1}$: dynamic range of the fixed point representation

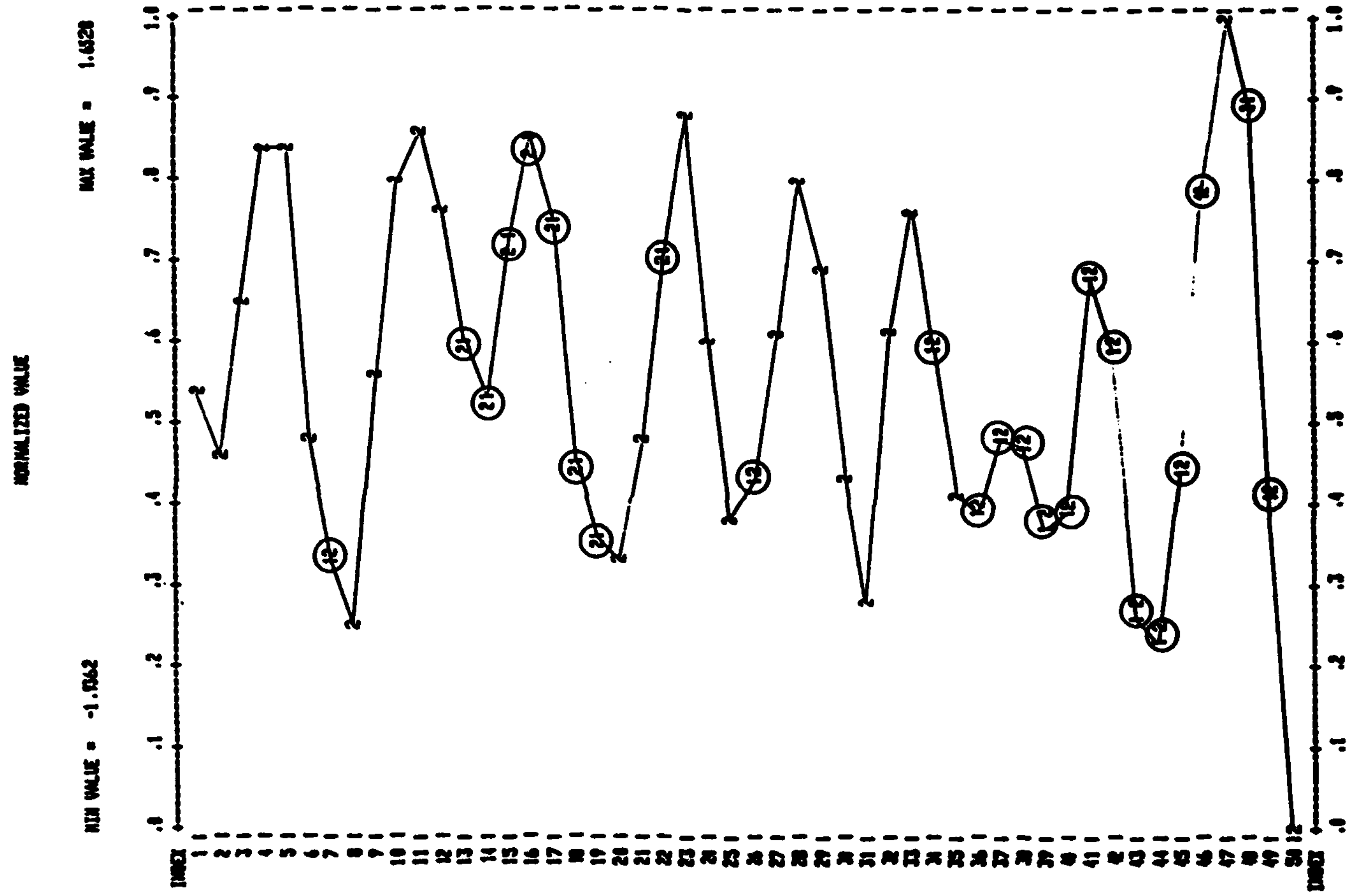
δ : design parameter that allows designer to control the probability of overflow

This formula generalizes the results of Mullis and Roberts to time varying cases.

The figure on the following page illustrates our experimental setup for implementing the Kalman filter on an 8086 microprocessor. The figure on the next page shows a typical simulation showing performance on the fixed point machine with that achievable on a floating point machine. The results apply to one-step prediction. The circles highlight places where the fixed point and floating point results differ by more than 1 bit in 6.



EXPERIMENTAL SET UP



PREDICTIONS USING FLOATING POINT AND FIXED POINT ARITHMETIC

PUBLICATIONS SUPPORTED BY THIS GRANT

Papers and Conference Presentations

"Modulo-2π Phase Sequence Estimation," IEEE Trans. Info. Theory, IT-26, pp 615-620, September 1980. (L. L. Scharf, D. D. Cox and C. J. Masreliez)

"Dynamic Programming for Phase and Frequency Tracking," NATO Advanced Study Institute, August 18-19, 1980, Copenhagen. (Louis Scharf). This invited paper subsequently published in L. Bjorno (Ed.), Underwater Acoustics and Signal Processing, D. Reidel Publishing Co.

"Exact Maximum Likelihood Identification of ARMA Models: A Signal Processing Perspective," EUSIPCO 80, European Signal Proc Conf, September 16-19, 1980, Lausanne, Switz. (Claude Gueguen and Louis Scharf) Invited Paper.

"Parametric Spectrum Modelling: A Signal Processing Perspective," First IEEE Workshop on Spectrum Analysis, August, 1981, Hamilton, Canada. (L. L. Scharf, C. J. Gueguen, and J.-P. Dugre) Invited Paper.

"Exact Likelihood Function for Vector ARMA processes," Colloq. on Fast Algorithms for Linear Dynamical Systems, September 21-25, 1981, Aussois, France (J.-P. Dugre, L. L. Scharf, and C. J. Gueguen)

"A Dynamic Programming Algorithm for Phase Estimation and Data Decoding on Random Phase Channels," IEEE Trans. Inform. Theory, IT-27, pp 581-595, September 1981. (O. Macchi and L. L. Scharf)

"Aspects of Dynamic Programming in Signal and Image Processing," IEEE Trans. on Autom. Contr., AC-26, pp 1018-1029, October 1981. (L. L. Scharf and H. Elliott) Bellman Special Issue.

"Linear Transformations and Parametric Spectrum Analysis," Proc. IEEE Intern. Conf. on Acoustics, Speech, and Signal Processing, Paris, France, May 3-5, 1982. (L. L. Scharf, et al.)

Technical Reports

"Parametric Spectrum Analysis of Stationary Random Sequences," Ph.D. Dissertation, Colorado State University, Ft. Collins, Colorado, J.-P. Dugre (9181).

"Frequency Estimation in Signal Plus Noise Models," M.S. Thesis, Colorado State University, Ft. Collins, Colorado, H. M. Anderson (1980)

"Dynamic Programming Algorithms for Frequency Tracking," M.S. Thesis, Colorado State University, Ft. Collins, Colorado, Marwan Orfali (1981).

"Fast Kalman Filtering for ARMA Processes: Fixed Point Implementation," M.S. Thesis, Colorado State University, Ft. Collins, Colorado, Sigurdur Sigurdsson (1982).

"Dynamic Programming for Time Delay Estimation," M.S. Thesis, Colorado State University, Ft. Collins, Colorado, Gregory Fetzer (1982).

SCIENTIFIC PERSONNEL EARNING ADVANCED DEGREES WHILE EMPLOYED ON PROJECT

Of the graduate students supported in full or in part under this grant, one has earned a Ph.D. and four have earned M.S. Degrees.

Ph.D.

J.-P. Dugre (1981), "Parametric Spectrum Analysis of Stationary Random Sequences," Ph.D. Dissertation, Colorado State University, Ft. Collins, Colorado.

M.S.

H. M. Anderson (9180), "Frequency Estimation in Signal Plus Noise Models," M.S. Thesis, Colorado State University, Ft. Collins, Colorado.

Marwan Orfali (1981), "Dynamic Programming Algorithms for Frequency Tracking," M.S. Thesis, Colorado State University, Ft. Collins, Colorado.

Sigurdur Sigurdsson (1982), "Fast Kalman Filtering for ARMA Processes: Fixed Point Implementation," M. S. Thesis, Colorado State University, Ft. Collins, Colorado.

Gregory Fetzer (1982), "Dynamic Programming for Time Delay Estimation," M.S. Thesis, Colorado State University, Ft. Collins, Colorado.

BIBLIOGRAPHY

1. L.L. Scharf, D.D. Cox, and C.J. Masreliez, "Modulo 2π Phase Sequence Estimation," IEEE Trans Inform Th, IT-26, Sept 1980

2. O. Macchi and L.L. Scharf," Dynamic Programming Algorithm for Phase Estimation and Data Decoding on Random Phase Channels," IEEE Trans Inform Th, IT-27, Sept 1981

3. L.L. Scharf and H. Elliott, "Aspects of Dynamic Programming in Signal and Image Processing," IEEE Trans Autom Contr, AC-26, Oct 1981

4. C.J. Gueguen and L.L. Scharf, "Exact Maximum Likelihood Identification of ARMA Models: A Signal Processing Perspective," EUSIPCO 80, Sept 1980, Lausanne, Switz

5. J.-P. Dugre, L.L. Scharf, and C.J. Gueguen, "Exact Likelihood Function for Vector ARMA Processes," Colloq on Fast Algorithms for Linear Dynamical Systems, Sept 21-25, 1981, Aussois, France

6. S. Sigurdsson, "Fast Kalman Filtering for ARMA Processes: Fixed Point Implementation," M.S. Thesis, Colorado State University, Ft. Collins, 1982

APPENDIX A : Reprints of Major Publications

This appendix contains selected reprints of papers published with ARO sponsorship under DAAG 29 79 C 0176.

Modulo-2π Phase Sequence Estimation

LOUIS L. SCHARF, SENIOR MEMBER, IEEE, DENNIS D. COX,
AND C. JOHAN MASRELIEZ, MEMBER, IEEE

Abstract—The probabilistic evolution of random walk on the circle is studied, and the results are used to derive a maximum *a posteriori* probability (MAP) *sequence* estimator for phase. The *sequence* estimator is a Viterbi tracker for tracking phase on a finite-dimensional grid in $[-\pi,\pi)$. The algorithm is shown to provide a convenient method for obtaining fixed-lag phase estimates. Performance characteristics are presented and compared with several published nonlinear filtering algorithms.

I. Introduction

Phase tracking is the classic nonlinear filtering problem. It arises in narrowband analog communication, data transmission, and spread spectrum communication. As usually stated, the problem is to obtain a causal estimate of the phase based on noisy phase-modulated observations. The best known solutions are phase-locked loops (PLL's).

In any truly nonlinear filtering approach to optimum phase tracking, the basic problem is to propagate the *a posteriori* density of the phase, conditioned on an increasing measurement record, much as is done in Kalman filtering. Unfortunately, there exist no finite-dimensional schemes for propagating the exact conditional density or for propagating a finite-dimensional sufficient statistic. One must approximate. The interested reader may consult [18] for a review of the best known techniques or, better yet, go directly to the appropriate source [1]–[13].

In this correspondence we propose an approach to phase *sequence* estimation (emphasis on the word sequence) that has its logical antecedents in the filtering philosophy of Youla [2] and the data decoding philosophy of Viterbi [14]. We pose a maximum *a posteriori* probability (MAP) sequence estimation problem that leads to nonlinear MAP equations not unlike the continuous-time MAP interval equations. Fortunately there exists a dynamic programming algorithm to efficiently solve for survivor phase sequences that approximate the desired MAP *sequence*. The algorithm also provides a handy mechanism for generating fixed-lag phase estimates, although this is not the problem for which the algorithm is derived. As is common in most of the current communications literature we call our dynamic programming algorithm a Viterbi algorithm.

Cahn [15] has suggested that phase may be tracked with delay in order to extend the so-called threshold. He proposes a Viterbi-like algorithm for tracking carrier phase sequences whose realizations satisfy dynamics constraints. There is certainly a philosophical link between Cahn's work and ours. In fact it was Cahn's paper that first aroused our interest in phase *sequence* estimation. However the approaches are really quite different. Ungerboeck [16] has proposed an algorithm for phase tracking that makes use of a delta-modulation approximation to the phase sequence and an approximate version of the Viterbi algorithm. Tufts and Francis [17] have also recently proposed an algorithm for obtaining smoothed phase estimates.

II. The Basic Problem

Let $\{Z_k\}$ denote the complex observation sequence

$$Z_k = e^{j\Phi_k} + N_k, \qquad k = 1, 2, \cdots,$$

where

$$\begin{aligned} N_k = U_k + jV_k, &\qquad N_k \perp\!\!\!\perp N_l \quad \text{for } k \neq l, \\ U_k : \mathfrak{N}(0,\sigma_n^2), &\qquad V_k : \mathfrak{N}(0,\sigma_n^2), \\ U_k \perp\!\!\!\perp V_l &\qquad \text{for all } k, l, \end{aligned} \tag{1}$$

$\{\Phi_k\}$ is a discrete-time phase sequence to be discussed shortly and $\{N_k\}$ is an additive noise sequence of independent identically distributed (i.i.d.) normal random variables. Our notation is that $N_k \perp\!\!\!\perp N_l$ means N_k and N_l are independent, and $U_k : N(0, \sigma_n^2)$ indicates that U_k is a normal random variable with mean 0 and variance σ_n^2. The sequence $\{Z_k\}$ may be thought of as a complex representation for the sample values appearing at the output of a quadrature demodulator. The problem is to estimate a realization of the entire *sequence* $\{\Phi_k\}_1^K$, say $\{\hat{\phi}_k\}_1^K$, from the measurement record $\{z_k\}_1^K$. It turns out that this formulation also provides a convenient way to generate a sequence of fixed-lag estimates. However, we emphasize that the basic problem under investigation is one of estimating an entire *sequence*, not one of generating a *sequence* of fixed-lag, fixed-interval, or fixed-point smoothing solutions. We make the obvious but important observation that the signal model (1) is invariant under a modulo-2π transformation on the phase.

III. Random Walk on the Circle as a Model for Phase Noise

The first, seemingly natural, choice for a random phase model is the Wiener process $W(t)$ with incremental variance σ_0^2. This is the most commonly used model for random phase acquisition. Most of the results in this correspondence may be obtained in a formal way using this phase model, but certain technical difficulties arise. First, there is no stationary distribution and, second, there is no rigorous way of defining a unique conditional probability for transitions from a modulo-2π value of $W(t)$ to another modulo-2π value at a later time $t+\tau$. The latter difficulty is particularly troublesome as one of the crucial parts of our modulo-2π phase sequence estimator is a transition probability matrix that characterizes phase transitions between modulo-2π values. By modeling phase as a random walk on the circle we avoid these technical difficulties. Other authors (see for example [7]) have also noted that the circle is the appropriate domain on which to study modulo-2π type sequences.

Let $\Phi(t)$ be a random walk on the circle, taking values in $[-\pi,\pi)$. Denote by the function $p(\phi_t/\phi_s)$ the conditional density of a transition from the value $\Phi(s)=\phi_s$ at time s to the value $\Phi(t)=\phi_t$ at time $t>s$. This conditional density satisfies the

Manuscript received May 1, 1978; revised July 15, 1979. This work was supported in part by the Office of Naval Research, Statistics and Probability Branch, Arlington, VA, under Contract N00014-75-C-0518 and by the Army Research Office, Research Triangle Park, NC, under Contract DAAG-29-79-C-0176.

L. L. Scharf is with the Electrical Engineering Department, Colorado State University, Ft. Collins, CO 80523

D. D. Cox is with the Department of Mathematics, University of Washington, Seattle, WA 98195 and the Marine Systems Center, Honeywell, Inc., Shilshole Avenue N.W., Seattle, WA 98195.

C. J. Masreliez is with the Marine Systems Center, Honeywell, Inc., Shilshole Avenue N.W., Seattle, WA 98195 and the Electrical Engineering Department, University of Washington, Seattle, WA 98195

0018-9448/80/0900-0615$00.75 ©1980 IEEE

partial differential equation [20]

$$\frac{\partial}{\partial t}p(\phi_t/\phi_s)=\frac{1}{2}\sigma_0^2\frac{\partial^2}{\partial\phi_t^2}p(\phi_t/\phi_s) \tag{2}$$

where σ_0^2 is the *infinitessimal* variance. This equation holds in the strip $-\pi<\phi_t<\pi$, $t>s$, for any fixed $-\pi<\phi_s<\pi$. The boundary conditions are

$$\lim_{t\to s}p(\phi_t/\phi_s)=\delta(\phi_t-\phi_s),$$
$$p(\phi_t=-\pi/\phi_s)=p(\phi_t=\pi/\phi_s),$$
$$\frac{\partial}{\partial\phi_t}p(\phi_t=-\pi/\phi_s)=\frac{\partial}{\partial\phi_t}p(\phi_t=\pi/\phi_s), \tag{3}$$

where δ is the Dirac delta function. We are using the convention that $\Phi(t)$ denotes a random variable and ϕ_t a realization. When the context is clear and there is no danger of confusion we will sometimes make no distinction in notation between a random variable and its realizations. The same cautionary note holds for the discrete-time random variable Φ_k and the realization ϕ_k.

The solution for $p(\phi_t/\phi_s)$ is

$$p(\phi_t/\phi_s)=\frac{1}{\sqrt{2\pi\sigma_0^2(t-s)}}\cdot\sum_{n=-\infty}^{\infty}\exp\left\{-\frac{1}{2\sigma_0^2(t-s)}(\phi_t-\phi_s-n2\pi)^2\right\}. \tag{4}$$

It is easily seen that the process $\Phi(t)$ is conditionally approximately $\mathfrak{N}(\phi_s,\sigma_0^2(t-s))$, given $\Phi(s)=\phi_s$ for small $t-s$. An eigenfunction expansion of the following form is also useful:

$$p(\phi_t/\phi_s)=\frac{1}{2\pi}\sum_{n=-\infty}^{\infty}\exp(-n^2\sigma_0^2(t-s)/2)\exp(jn(\phi_t-\phi_s)). \tag{5}$$

This is simply Poisson's summation formula for (4). From this expression it is clear that $\Phi(t)$ becomes uniformly distributed as $t-s\to\infty$. Equation (5) has also been noted in [7].

Consider the discrete-time sequence $\{\Phi_k\}$ obtained by sampling $\Phi(t)$ at the periodic sampling instants $t=kT$, $k=0,1,\cdots$. Call ϕ_k a realization of Φ_k. The transition density from ϕ_{k-1} to ϕ_k is found from (4) with $\sigma_w^2=\sigma_0^2T$ to be

$$p(\phi_k/\phi_{k-1})=\frac{1}{\sqrt{2\pi\sigma_w^2}}\sum_{n=-\infty}^{\infty}\exp\left\{-\frac{1}{2\sigma_w^2}(\phi_k-\phi_{k-1}-n2\pi)^2\right\}. \tag{6}$$

By the Markov property of $\Phi(t)$ it follows that $\{\Phi_k\}$ is a Markov sequence for which the joint distribution of $\{\Phi_k\}_1^K$ may be written

$$p(\{\phi_k\}_1^K)=\prod_{k=1}^{K}p(\phi_k/\phi_{k-1}), \tag{7}$$

where

$$p(\phi_1/\phi_0):U[-\pi,\pi),$$

and the notation $p(\phi_1/\phi_0):U[-\pi,\pi)$ indicates that Φ_1 is uniformily distributed on $[-\pi,\pi)$. Other choices are also admissable: for example, $p(\phi_1/\phi_0)=\delta(\phi_1-\phi_0)$ with ϕ_0 known corresponds to a given initial phase.

One may obtain the same discrete-time model for $\{\Phi_k\}$ by considering Φ_k to be a modulo-2π version of the following discrete-time random walk:

$$\Theta_k=\Theta_{k-1}+W_k,\qquad W_k\perp\!\!\!\perp W_l \text{ for } k\neq l,$$
$$W_k:\mathfrak{N}(0,\sigma_W^2),\qquad \sigma_W^2=\sigma_0^2T. \tag{8}$$

The modulo-2π version of Θ_k, call it $\bar\Theta_k$, may be written

$$\bar\Theta_k=\tilde\Theta_k,\qquad \tilde\Theta_k=\bar\Theta_{k-1}+W_k. \tag{9}$$

Given $\bar\Theta_{k-1}=\bar\theta_{k-1}$, the random variable $\tilde\Theta_k$ is $\mathfrak{N}(\bar\theta_{k-1},\sigma_W^2)$. As $\bar\Theta_k$ is a modulo-2π version of $\tilde\Theta_k$ it follows that the conditional density of $\bar\Theta_k$, given $\bar\Theta_{k-1}=\bar\theta_{k-1}$, is the folded normal density of (6). For this reason we will often call the discrete-time process on the circle $\{\Phi_k\}$ a modulo-2π version of the discrete-time random walk $\{\Theta_k\}$.

In Section VII we discretize the phase space $[-\pi,\pi)$ to phase values ξ_m, $m=0,\cdots,M-1$ with M odd. It is then necessary to characterize the transition probability from ξ_l to ξ_m for all M^2 pairs of (ξ_l,ξ_m). We choose for our definition of this transition probability

$$\bar p(\phi_k=\xi_m/\phi_{k-1}=\xi_l)=b_lp(\phi_k=\xi_m/\phi_{k-1}=\xi_l) \tag{10}$$

with b_l selected so that

$$\sum_{m=0}^{M-1}\bar p(\phi_k=\xi_m/\phi_{k-1}=\xi_l)=1,\qquad l=0,1,\cdots,M-1. \tag{11}$$

The sum on n in (6) must, of course, be truncated. This truncation may be selected to give the desired accuracy before the algorithm of Section VII is run. There is no series truncation whatsoever in the algorithm itself.

There is an important symmetry property of (10). If the ξ_l are equally spaced points on $[-\pi,\pi)$, for example $\xi_l=l2\pi/M-(M-1)\pi/M$, the function $\bar p$ depends only on $|\xi_m-\xi_l|$. Thus if the values of (10) are organized into an $M\times M$ matrix of transition probabilities, the matrix is Toeplitz. We may compute the M-dimensional vector $\bar Q=(\bar q_0,\bar q_1,\cdots,\bar q_{M-1})$ with $\bar q_n=\bar p(\phi_k=\xi_0/\phi_{k-1}=\xi_n)$ and obtain any value of $\bar p(\phi_k=\xi_m/\phi_{k-1}=\xi_l)$ as $\bar q_n$ with $n=|m-l|$. In this way only an M-vector of transition probabilities need be stored for cyclic reading.

IV. The MAP Sequence Estimation Problem for Modulo-2π Phase

Consider the following maximization with respect to the modulo-2π phase *sequence* $\{\phi_k\}_1^K$:

$$\max_{\{\phi_k\}_1^K}p(\{z_k\}_1^K,\{\phi_k\}_1^K), \tag{12}$$

where $p(\cdot,\cdot)$ is the joint density function for the K measurements $\{z_k\}_1^K$ and the K modulo-2π phase values $\{\phi_k\}_1^K$. Maximization of this joint density function is equivalent to maximization of the *a posteriori* density $p(\{\phi_k\}_1^K/\{z_k\}_1^K$. The joint density in (12) may be written

$$p(\{z_k\}_1^K,\{\phi_k\}_1^K)=p(\{z_k\}_1^K/\{\phi_k\}_1^K)p(\{\phi_k\}_1^K)$$
$$=p(\{\phi_k\}_1^K)\prod_{k=1}^{K}\mathfrak{N}_{z_k}(e^{j\phi_k},\sigma_n^2), \tag{13}$$

where the last line follows since the N_k in (1) are i.i.d. normal random variables and because, conditionally,

$$p(z_k/\phi_k):\mathfrak{N}_{z_k}(e^{j\phi_k},\sigma_n^2). \tag{14}$$

Here $\mathfrak{N}_{z_k}(e^{j\phi_k},\sigma_n^2)$ indicates that the conditional density of the complex random variable Z_k (conditioned on ϕ_k) is normal with mean $e^{j\phi_k}$ and variance σ_n^2:

$$N_{z_k}(e^{j\phi_k},\sigma_n^2)=(2\pi\sigma_n^2)^{-1}\exp\left\{-\frac{1}{2\sigma_n^2}|z_k-e^{j\phi_k}|^2\right\}. \tag{15}$$

Dropping phase independent terms we may write the MAP sequence estimation problem as

$$\max_{\{\phi_k\}_1^K}\Gamma_K$$

where

$$\Gamma_K = \frac{1}{\sigma_n^2} \mathrm{Re} \sum_{k=1}^{K} \Lambda_k + \sum_{k=1}^{K} \ln p(\phi_k/\phi_{k-1}), \tag{16}$$

Λ_K is the phase-corrected vector

$$\Lambda_K = \sum_{k=1}^{K} c_k e^{j(\psi_k - \phi_k)} = \sum_{k=1}^{K} z_k e^{-j\phi_k}, \tag{17}$$

and c_k and ψ_k are, respectively, envelope and phase variables: $z_k = c_k e^{j\psi_k}$, $\psi_k \in [-\pi, \pi]$, $c_k \in [0, \infty)$. It is clear from this form that the MAP phase sequence will be one that stays reasonably close to the noisy phase variables ψ_k (to make $\cos(\psi_k - \phi_k)$ large) while also maintaining a trajectory that is *a priori* reasonably likely. Thus the MAP sequence strikes a balance between what the noisy data ψ_k *says* the phase is doing and what the transition probabilities $p(\phi_k/\phi_{k-1})$ say the phase *can* do. When the envelope c_k is large there is more of a tendency to believe the measured ψ_k. This curious effect may be explained by noting that the phase statistic Ψ_k is a modulo-2π unbiased estimate of ϕ_k with a variance that decreases approximately inversely with increasing c_k [18].

V. Characteristics of the MAP Sequence

Given the envelope and phase variables $\{c_k\}_1^K$ and $\{\psi_k\}_1^K$, the MAP phase sequence $\{\hat{\phi}_k\}_1^K$ may be obtained by equating the derivatives of Γ_K to zero:

$$\frac{\partial}{\partial \hat{\phi}_k} \ln p(\hat{\phi}_k/\hat{\phi}_{k-1}) + \frac{\partial}{\partial \hat{\phi}_k} \ln p(\hat{\phi}_{k+1}/\hat{\phi}_k) + \frac{1}{\sigma_n^2} I_m(\Lambda_k - \Lambda_{k-1}) = 0, \quad k = 1, 2, \cdots, K. \tag{18}$$

The boundary conditions are $\Gamma_0 = 0$ and

$$\begin{aligned} &1) \quad \frac{\partial}{\partial \hat{\phi}_1} \ln p(\hat{\phi}_1/\hat{\phi}_0) = 0, \\ &2) \quad \frac{\partial}{\partial \hat{\phi}_K} \ln p(\hat{\phi}_{K+1}/\hat{\phi}_K) = 0. \end{aligned} \tag{19}$$

Condition 1) simply reflects the fact that $p(\phi_1/\phi_0)$ is uniform on $[-\pi, \pi)$. Condition 2) is a mathematical convenience that allows us to put all the equations of (18) in the same form. Of course $\hat{\phi}_0$ and $\hat{\phi}_{K+1}$ are not computed from the data $\{c_k\}_1^K$ and $\{\psi_k\}_1^K$.

Equations (18) are nonlinear equations with two-point boundary conditions. They are analogous to the continuous-time MAP equations obtained for phase tracking on an interval. While we cannot solve the equations of (18) explicitly we can make some very interesting observations regarding the properties of the MAP phase sequence.

It is easily verified from the conditional density of (6) that

$$\frac{\partial}{\partial \phi_k} \ln p(\phi_{k+1}/\phi_k) = -\frac{\partial}{\partial \phi_{k+1}} \ln p(\phi_{k+1}/\phi_k). \tag{20}$$

Therefore when the K equations of (18) are summed and the boundary conditions applied, all terms involving $\ln p(\hat{\phi}_{k+1}/\hat{\phi}_k)$ cancel. The sum on the terms involving $\mathrm{Im}(\Lambda_k - \Lambda_{k-1})$ telescopes, and we are left with the result

$$\mathrm{Im}\, \hat{\Lambda}_K = 0. \tag{21}$$

Here $\hat{\Lambda}_K$ is Λ_K with ϕ_k set to the MAP estimate $\hat{\phi}_k$ for $k = 1, 2, \cdots, K$. This allows us to make the following observation: while maximizing the objective function Γ_K, the MAP sequence $\{\hat{\phi}_k\}_1^K$ yields a maximum value for Γ_K of

$$\hat{\Gamma}_K = \frac{1}{\sigma_n^2} \mathrm{Re}\, \hat{\Lambda}_K + \sum_{k=1}^{K} \ln p(\hat{\phi}_{k+1}/\hat{\phi}_k) \tag{22}$$

with the property that $\mathrm{Im}\, \hat{\Lambda}_K = 0$. This property is illustrated in Fig. 1. We note that there are many other sequences that satisfy the condition $\mathrm{Im}\, \hat{\Lambda}_K = 0$ (e.g., the sequence of maximum likelihood estimates $\hat{\phi}_k = \psi_k$), but these sequences do not also maximize Γ_K.

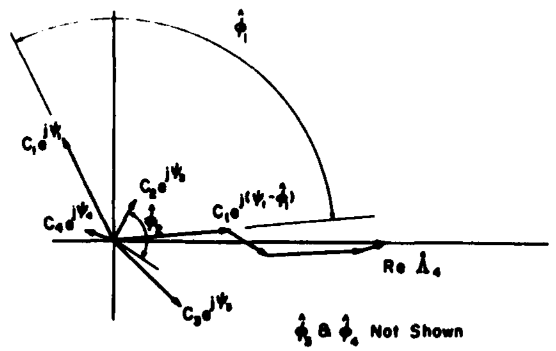


Fig. 1. Nature of the MAP sequence.

VI. The MAP Sequence for Fixed Phase Acquisition

Suppose the underlying phase sequence $\{\Phi_k\}_1^K$ is known to be a constant sequence with the value of the constant uniformly distributed on $[-\pi, \pi]$. In this case the MAP sequence estimate is identical with the maximum likelihood (ML) estimate of an unknown phase parameter ϕ in a complex normal model. For this reason, and for the insight it gives into phase estimation, we include in the following paragraphs a short discussion of constant phase and envelope models. The inclusion of an unknown envelope c generalizes the discussion without changing the nature of the phase estimate. This follows from the fact that the phase estimate is uncoupled from the envelope estimate. The converse is not true.

Consider the joint density function for the data $\{Z_k\}_1^K$, parameterized by the envelope c and the phase ϕ:

$$g(\{z_k\}_1^K) = \frac{1}{(2\pi\sigma_n^2)^K} \exp\left\{ -\frac{1}{2\sigma_n^2} \sum_{k=1}^{K} |z_k - ce^{j\phi}|^2 \right\} \tag{23}$$

$$= d(\phi, c) h(\{z_k\}_1^K) \exp\left\{ \frac{1}{\sigma_n^2} \mathrm{Re}\, ce^{-j\phi} \sum_{k=1}^{K} z_k \right\} \tag{24}$$

where

$$d(\phi, c) = \frac{1}{(2\pi\sigma_n^2)^K} \exp\left\{ -\frac{1}{2\sigma_n^2} Kc^2 \right\}, \tag{25}$$

$$h(\{z_k\}_1^K) = \exp\left\{ -\frac{1}{2\sigma_n^2} \sum_{k=1}^{K} |z_k|^2 \right\}.$$

It follows from the factorization theorem [21, p. 115] that the complex statistic $K^{-1}\Sigma_{k=1}^K Z_k$ is sufficient for the parameter pair (c, ϕ). The ML estimate for the composite parameter $a \triangleq ce^{j\phi}$ is

$$\hat{a} = K^{-1} \sum_{k=1}^{K} z_k. \tag{26}$$

This ML estimator is consistent, unbiased, efficient, and minimum variance unbiased. The corresponding ML estimates for $\hat{c}$ and $\hat{\phi}$ are

$$\hat{c} = K^{-1} \left| \sum_{k=1}^{K} z_k \right|,$$

$$\hat{\phi} = \arg \sum_{k=1}^{K} z_k. \tag{27}$$

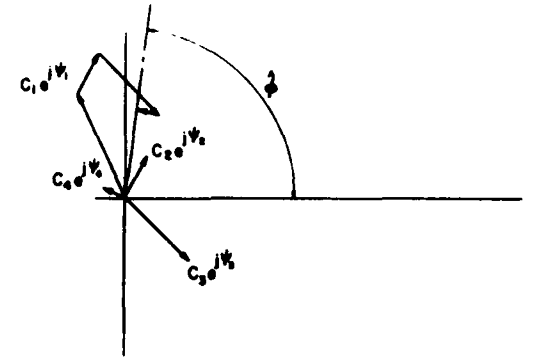


Fig. 2. MAP estimate for fixed phase acquisition.

Let $\hat{C}$ and $\hat{\Phi}$ be the estimators corresponding to the estimates $\hat{c}$ and $\hat{\phi}$. The estimator $\hat{C}$ is consistent, unbiased, efficient, and minimum variance unbiased. The phase estimator $\hat{\Phi}$ is not efficient and no efficient estimator exists. It is consistent but biased. However it is modulo-2π unbiased, which is the property we want. The phase estimate $\hat{\phi}$, also obtained in [8] and [17], in different ways, is illustrated in Fig. 2.

Define the modulo-2π estimator error $\Delta\hat{\Phi}=(\hat{\Phi}-\phi)\text{mod } 2\pi$. We may write

$$K^{-1}\sum_{k=1}^{K} Z_k e^{-j\phi}=\hat{C}e^{j(\hat{\Phi}-\phi)}=\hat{C}e^{j\Delta\hat{\Phi}}. \tag{28}$$

The statistic $K^{-1}\Sigma_{k=1}^{K} Z_k e^{-j\phi}$ is $\mathfrak{N}(c,\sigma_n^2/K)$. The Jacobian of the transformation between $(C, \Delta\hat{\Phi})$ and $K^{-1}\Sigma_{k=1}^{K} Z_k e^{-j\phi}$ is $\hat{C}$. Therefore the joint density of $\hat{C}$ and $\Delta\hat{\Phi}$ is

$$\begin{aligned} g(\hat{c},\Delta\hat{\phi}) &= \frac{\hat{c}}{2\pi\sigma_n^2/K}\exp\left\{-\frac{1}{2\sigma_n^2/K}|\hat{c}e^{j\Delta\hat{\phi}}-c|^2\right\} \\ &= \frac{\hat{c}}{2\pi\sigma_n^2/K}\exp-\frac{1}{2\sigma_n^2/K}\left[\hat{c}^2-2c\hat{c}\cos(\Delta\hat{\phi})+c^2\right]\Bigg\}. \end{aligned} \tag{29}$$

This r sult is equivalent to [22, eq. (9.46), p. 413] with appropriate change of notation. In (29) it is assumed that $-\pi<\Delta\hat{\phi}<\pi$. On this interval $g(c,\Delta\hat{\phi})$ is symmetrical about zero and therefore unbiased. We emphasize that $\hat{\Phi}$ is only modulo-2π unbiased.

VII. The Viterbi Algorithm

The MAP sequence estimation problem is stated in (18). Note that Γ_k satisfies the recursion

$$\begin{aligned} \Gamma_k &= \Gamma_{k-1}+\frac{1}{\sigma_n^2}c_k\cos(\psi_k-\hat{\phi}_k)+\ln p(\phi_k/\phi_{k-1}), \\ \Gamma_1 &= \frac{1}{\sigma_n^2}c_1\cos(\psi_1-\phi_1)+\ln p(\phi_1/\phi_0). \end{aligned} \tag{30}$$

The so-called path metric is

$$\frac{1}{\sigma_n^2}c_k\cos(\psi_k-\hat{\phi}_k)+\ln p(\phi_k/\phi_{k-1}). \tag{31}$$

The maximization problem for obtaining the MAP phase sequence may now be written

$$\max_{\{\phi_k\}_{K-1}^{K}}\left[\max_{\{\phi_k\}_1^{K-2}}\Gamma_{K-1}+\ln p(\phi_K/\phi_{K-1})+\frac{1}{\sigma_n^2}c_K\cos(\psi_K-\phi_K)\right]. \tag{32}$$

This form leads to the following observation: the maximizing trajectory (call it $\{\hat{\phi}_k\}_1^K$), passing through $\hat{\phi}_{K-1}$ on its way to $\hat{\phi}_K$, must arrive at $\hat{\phi}_{K-1}$ along a route $\{\hat{\phi}_k\}_1^{K-2}$ that maximizes Γ_{K-1}. For if it did not we could retain $\hat{\phi}_{K-1}$ and $\hat{\phi}_K$ and replace $\{\hat{\phi}_k\}_1^{K-2}$ with a different sequence to get a larger value for Γ_K. It is this observation which forms the basis of forward dynamic problem.

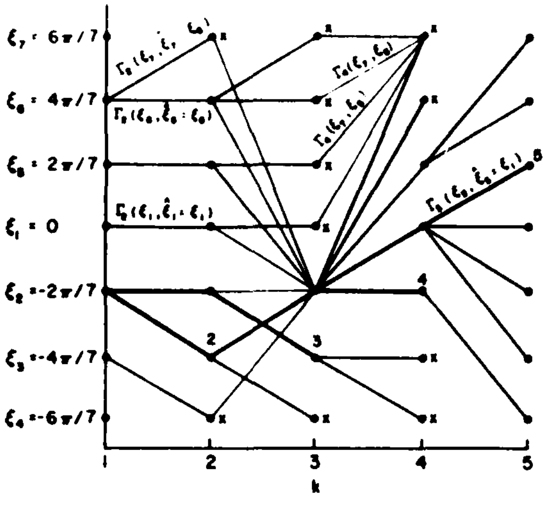


Fig. 3. Phase trellis illustrating evolution of surviving phase tracks.

The trellis of Fig. 3 illustrates how the maximization of (32) proceeds. Tabulated values of $p(\phi_k/\phi_{k-1})$ are stored in a square array (or vector which is read cyclically) whose dimensions depend upon how finely the interval $[-\pi,\pi)$ is discretized. Let $\Xi\times\Xi$, with $\Xi=\{\xi_l\}_1^M$, be the finite-dimensional grid for which $p(\phi_k/\phi_{k-1})$ is defined. That is, ϕ_k is assumed to take on only the values $\phi_k=\xi_l$, $l=1,2,\cdots,M$, for each k. Let $\Gamma_k(\xi_l,\xi_m)$ be the value of the metric Γ_k corresponding to a phase trajectory $\{\phi_j\}_1^k$ which terminates at phase-state ξ_l at stage k, after passing through stage ξ_m at stage $k-1$.

The algorithm begins with a computation of $\Gamma_1(\xi_l,\xi_l)$, $l=1,2,\cdots,M$ based on measured values of c_1 and ψ_1. If all phase values are equally likely *a priori*, then $\ln p(\phi_1/\phi_0)$ is constant on all values ξ_l. Otherwise there is some *a priori* weighting in favor of some of the ξ_l. A new measurement pair (c_2,ψ_2) is obtained at $k=2$ and $\Gamma_2(\xi_1,\xi_m)$ is computed for $m=1,2,\cdots,M$ using a table look-up (for example in a read-only memory) for the $p(\phi_2=\xi_1/\phi_1=\xi_m)$. The maximum value of $\Gamma_2(\xi_1,\xi_m)$ is determined (over all originating series ξ_m) and the corresponding sequence $(\xi_1,\hat{\xi}_1)$ is saved as a *survivor* sequence terminating at ξ_1 at stage 2; $\hat{\xi}_1$ denotes the originating state. The survivor sequence is labeled with its corresponding *length* $\Gamma_2(\xi_1,\hat{\xi}_1)$. This calculation is repeated for each possible value of phase until all pairs $(\xi_l,\hat{\xi}_l)$ and corresponding lengths $\Gamma_2(\xi_l,\hat{\xi}_l)$, $l=1,2,\cdots,M$, have been computed and stored (for example in a random access memory). There is a unique survivor sequence corresponding to each state ξ_l, $l=1,2,\cdots,M$. *Caution*: In the pair $(\xi_l,\hat{\xi}_l)$ the originating state $\hat{\xi}_l$ depends on ξ_l; i.e., $\hat{\xi}_l=\hat{\xi}_l(\xi_l)$. The measurements c_2 and ψ_2 may now be discarded along with all extinct sequences. A new measurement pair (c_3,ψ_3) is now obtained. and the procedure continues.

Let $(\hat{\phi}_1(k),\hat{\phi}_2(k),\cdots,\hat{\phi}_k(k))$ be the MAP sequence based on k measurements; this sequence has the maximum value of Γ_k. The parenthetical notation (k) denotes dependence on measurement interval. In general the MAP sequence estimate $(\hat{\phi}_1(k+1),\cdots,\hat{\phi}_{k+1}(k+1)$ based on measurements up to stage $k+1$ may differ from the previous sequence estimate at every stage from 1 to k. However, as a practical matter, one can choose a sufficiently large depth parameter k_0 so that the sequence of fixed-lag estimates

$$\hat{\phi}_{k-k_0}(k), \qquad k=k_0+1,k_0+2,\cdots \tag{33}$$

gives an approximate MAP sequence estimate. Here $\hat{\phi}_{k-k_0}(k)$ is simply the phase value k_0 stages back, in the MAP sequence estimate based on k measurements. In this way one obtains a phase track with delay k_0.

Following Forney [23] we may summarize the storage and computational requirements for the phase tracking algorithm as follows.

Storage

k	(time index),
$(\xi_l, \xi_l, \cdots, \cdots), l=1,2,\cdots,M$	(survivor phase sequence terminating in ξ_l at stage k),
$\Gamma_k(\xi_l, \hat{\xi}_l, \cdots), l=1,2,\cdots,M$	(survivor metric),
$p(\xi_l/\xi_m), l,m=1,\cdots,M$	(transition probability matrix).

Initialization

$$k=1,$$

$$\hat{\xi}_l=\xi_l, \qquad l=1,\cdots,M,$$

$$\Gamma_1(\xi_l,\hat{\xi}_l)=\ln p(\phi_1=\xi_l/\phi_0)+\frac{1}{2\sigma_n^2}2c_1\cos(\psi_1-\xi_l), \qquad l=1,2,\cdots,M.$$

Recursion

$$\Gamma_{k+1}(\xi_l,\xi_m)=\Gamma_k(\xi_m,\hat{\xi}_m)+\ln p(\phi_{k+1}=\xi_l/\phi_k=\xi_m)+\frac{1}{2\sigma_n^2}2c_{k+1}\cos(\psi_{k+1}-\xi_l), \qquad l=1,2,\cdots,M,$$

$$\min_{(\xi_m)}\Gamma_{k+1}(\xi_l,\xi_m), \qquad l=1,2,\cdots,M.$$

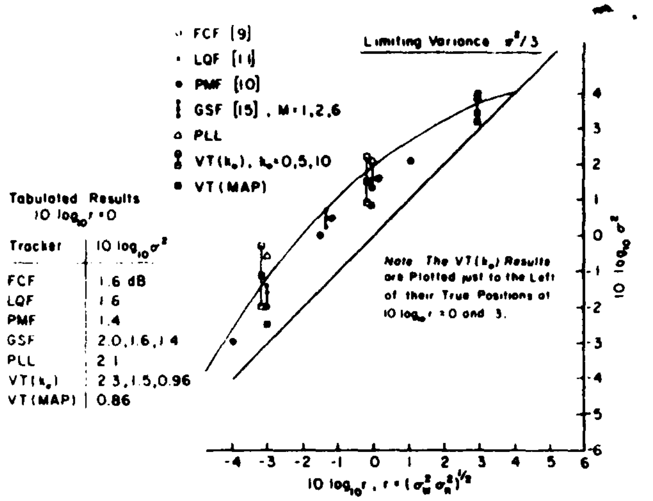
Measurement / Computation

c_k	(envelope),
ψ_k	(phase),
$c_k\cos(\xi_l-\psi_k)+\ln p(\phi_k=\xi_l/\phi_{k-1}=\xi_m)$	(path metric).

In Fig. 3 typical trajectories for this algorithm are illustrated. The heavy lines denote survivors and the light lines denote path metric calculations that are made and then discarded in favor of survivors. At the third stage all calculations $\Gamma_3(\xi_2,\xi_m)$ are illustrated with light lines; the heavy line from ξ_3 to ξ_2 illustrates that this path gives maximum $\Gamma_2(\xi_2,\xi_m)$ and is therefore labeled a survivor. Of course $\hat{\xi}_2=\xi_3$. The letters x on the trellis illustrate sequences that have survived for a while before being exterminated by the weight of evidence. The very heavy line at each measurement stage k denotes the current MAP sequence. The labeling numbers on the heavy paths denote the current MAP sequence. The sequence of end points labeled with the numbers is a sequence of *phase estimates*. Note this *sequence of phase estimates* differs from the *MAP phase sequence*. The latter, being a smoothing solution, is in fact generally smoother than the former.

VIII. Performance Results

The phase space $[-\pi,\pi)$ has been divided into $M=11$ equally spaced points and the Viterbi algorithm for phase tracking implemented as outlined in Section VII. The crucial conditional probabilities $p(\phi_k=\xi_m/\phi_{k-1}=\xi_l)$ have been computed as outlined in Section III and stored in an M vector for cyclic reading. Random phase trajectories and measurement variables have been generated according to (8) and (1). The results of several Monte-Carlo simulations are presented in Fig. 4. Each Monte-Carlo result has been obtained by running the Viterbi phase tracker (and the PLL) over 40 different trajectories, each trajectory beginning with a uniformly distributed phase variable at $k=1$ and continuing for 500 points. Various values of depth parameter k_0 have been used, as indicated in the figure. (See [8] for a discussion of corresponding statistical sampling errors and [18] for additional Monte-Carlo results.)



Tabulated Results
10 $\log_{10} r$ = 0

Tracker	10 $\log_{10}\sigma^2$
FCF	1.6 dB
LQF	1.6
PMF	1.4
GSF	2.0, 1.6, 1.4
PLL	2.1
VT(k_0)	2.3, 1.5, 0.96
VT(MAP)	0.86

Fig. 4. Performance results for $\sigma_w^2/\sigma_n^2=0.01$.

In Fig. 4 Monte-Carlo simulation results for the Viterbi tracker are presented for the parametrizations commonly considered in the literature. The results are compared with the point mass filter (PMF) [8], the Fourier coefficient filter (FCF) [7], the linear quadrature filter (LQF) [9], and the Gaussian sum filter (GSF) [13]. Also shown are our simulation results for the PLL. These results are presented to legitimize the simulation. The Viterbi tracker makes up more than 1.0 of the 2.0 dB performance gap between the PLL and an idealized linear tracker. In terms of rms phase error (in radians), the comparison between the PLL and the Viterbi tracker goes as follows. The PLL has an rms phase error of 1.26 rad at $r=1.0$. The maximum achievable percentage improvement is 21 percent, corresponding to an ideal filter with rms phase error of 1.0 rad. The rms error for the Viterbi tracker operating at $r=1.0$ with $k_0=10$ is 1.12. This represents an improvement of 11 percent over the PLL. The results for $k_0=0$ show that (as expected) the Viterbi tracker is not as good as a PLL as a zero-lag filter. In Fig. 4 the heavy squares denoted by VT(MAP) (see the symbol key) correspond to the smoothing variance achieved when the MAP sequence for a 500 sample run is used as the phase estimate. The results are averaged over 40 such runs. The tabulated results in Fig. 4 summarize the performance characteristics of many different nonlinear phase trackers. In the figure, performance results for the Viterbi tracker are plotted just to the left of their true positions to avoid cluttering the presentation.

We hasten to emphasize in the interest of fair play that all results presented here for nonzero k_0 are in reality smoothing solutions. Such solutions are expected to deliver the usual smoothing gains over filtering solutions. This does not detract from the Viterbi tracker as an attractive alternative in those applications where a short delay may be accepted in exchange for 1–2 dB performance gains.

IX. Conclusions

We have derived a Viterbi algorithm for obtaining approximate MAP phase sequence estimates on $[-\pi,\pi)$. The algorithm is simple and fast by nonlinear filtering standards and ideally organized for hardware implementation. More dramatic performance gains than those illustrated in Fig. 4 may be achieved when phase fluctuations are severe, i.e., when $\sigma_w^2/\sigma_n^2 > 0.01$. The reader is referred to [19] for applications of these results to phase coherent data communication.

Acknowledgment

The authors acknowledge the support of J. Lord, R. McGough, and B. Picinbobo. They thank O. Macchi for helpful discussions and for her critique of the manuscript. C. Pariente conducted the Monte-Carlo simulations at the University of Paris-Sud using software originally developed C. J. Masreliez.

References

[1] F. Lehan and R. Parks, "Optimum demodulation," *IRE National Conv. Rec.*, pt. 8, pp. 101–103, 1953.

[2] D. C. Youla, "The use of maximum likelihood in estimating continuously modulated intelligence which has been corrupted by noise," *IRE Trans. Inform. Theory*, vol. IT-3, pp. 90–105, Mar. 1954.

[3] D. M. Detchmendy and R. Sridhar, "Sequential estimation of states and parameters in noisy dynamical systems," *ASME J. Basic Eng.*, pp. 362–368, June 1966.

[4] A. B. Baggeroer, "Nonlinear MAP interval estimation," Res. Lab. Electron., MIT, Cambridge, MA, QPR no. 85, pp. 249–253, Apr. 1967.

[5] H. J. Kushner, "On the differential equations satisfied by conditional probability densities of Markov processes, with applications," *J. SIAM Contr.*, Ser. A, vol. 2, pp. 106–119, 1964.

[6] D. Snyder, *The State Variable Approach to Continuous Estimation with Applications to Analog Communication Theory*. Cambridge, MA: MIT, 1969.

[7] A. S. Willsky, "Fourier series and estimation on the circle with applications to synchronous communication-Part I: Analysis," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 577–583, Sept. 1974.

[8] R. S. Bucy and A. J. Mallinckrodt, "An optimal phase demodulator," *Stochastics*, vol. 1, pp. 3–23, 1973.

[9] D. E. Gustafson and J. L. Speyer, "Linear minimum variance filters applied to carrier tracking," *IEEE Trans. Automat. Contr.*, vol. AC-21, pp. 65–73, Feb. 1976.

[10] C. N. Kelly and S. C. Gupta, "The digital phase-locked loop as a near-optimum FM demodulator," *IEEE Trans. Commun.*, vol. COM-20, pp. 406–411, June 1972.

[11] R. S. Bucy and H. Youssef, "Fourier realization of the optimal phase demodulator," in *Proc. Symp. Nonlinear Estimation and Its Applications*, San Diego, CA, 1973.

[12] H. W. Sorenson and D. L. Alspach, "Recursive Bayesian estimation using Gaussian sums," *Automat.*, vol. 7, pp. 465–479, 1971.

[13] P. K. S. Tam and J. B. Moore, "A Gaussian sum approach to phase and frequency estimation," *IEEE Trans. Commun.*, vol. COM-25, pp. 935–942, Sept. 1977.

[14] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inform. Theory*, vol. IT-15, pp. 260–269, Apr. 1969.

[15] C. R. Cahn, "Phase tracking and demodulation with delay," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 50–58, Jan. 1974.

[16] G. Ungerboeck, "New applications for the Viterbi algorithm: Carrier phase tracking in synchronous data-transmission systems," in *Proc. Nat. Telecommun. Conf.*, 1974.

[17] D. W. Tufts and J. T. Lewis, "Estimation and tracking of parameters of narrowband signals by iterative processing," *IEEE Trans. Inform. Theory*, vol. IT-23, pp. 742–751, Nov. 1977.

[18] L. L. Scharf, D. D. Cox, and C. J. Masreliez, "Modulo-2π phase sequence estimation," ONR Tech. Rep. #27, Feb. 1978.

[19] L. L. Scharf, "A Viterbi algorithm for modulo-2π phase tracking in coherent data communication systems," ONR Tech. Rep. #25, Dec. 1977.

[20] W. Feller, *Introduction to Probability Theory and Its Applications*, vol. II, 2nd ed. New York: Wiley, 1966.

[21] T. S. Ferguson, *Mathematical Statistics: A Decision Theoretic Approach*. New York: Academic, 1967.

[22] D. Middleton, *An Introduction to Statistical Communication Theory*. New York: McGraw-Hill, 1960.

[23] G. D. Forney, "The Viterbi algorithm," *Proc. IEEE*, pp. 268–278, Mar. 1973.

DYNAMIC PROGRAMMING FOR PHASE AND FREQUENCY TRACKING

Louis L. Scharf

Electrical Engineering Department*
Colorado State University
Fort Collins, CO 80523 USA

Abstract

The techniques of dynamic programming have found a variety of successful applications in signal and system theory. In this paper we show how two knotty nonlinear filtering problems--phase and frequency tracking--may be formulated and solved as forward dynamic programming problems. The resulting solutions are fixed interval smooths in which a most likely sequence is passed through a data record.

1. INTRODUCTION

Phase and frequency tracking problems comprise some of the most nettlesome nonlinear filtering problems in the realm of signal processing. These problems have held the interest of control and communication theorists at least since 1953/54 when Lehan and Parks [1] and Youla [2] published their work on maximum likelihood and optimum demodulation on an interval. Over the years Cox [3], Viterbi [4], Cahn [5], Forney [6], and a host of others have advocated dynamic programming for the solution of nonlinear filtering problems. This is a paper in the same tradition.

In this paper we discuss forward dynamic programming as a

*This work supported by the Army Research Office, Research Triangle Park, NC under contract DAAG29-79-C-0176 and by the Office of Naval Research, Arlington, VA under contract N00014-75-C-0518.

Bjørnø (ed.), Underwater Acoustics and Signal Processing
Copyright © 1981 by D. Reidel Publishing Company

technique for finding the maximum a posteriori (MAP) phase or frequency modulated sequence to pass through a data set. The key idea is to pose a Markov chain model on the circle $[0,2\pi)$ for phase or frequency, and then generate candidate MAP sequences that are consistent with the data and the a priori probability structure. More details may be found in [7] & [8].

2. PHASE SEQUENCE ESTIMATION

Figures 1 & 2 depict two classical phase estimation problems: constant phase estimation and random walk phase estimation. In these figures and throughout the paper $Z_t = \{z_i,\ldots,z_t\}$ denotes the data set and the $n_k, k=i,i+1,\ldots,t$ are complex i.i.d. $N(0,\sigma^2)$ r.v.s. A Markov transition density (or probability mass function) is denoted $p(\cdot/\cdot)$; $f(\cdot,\cdot,\ldots,\cdot)$ denotes a joint density function; $[\cdot]$ denotes integer part.

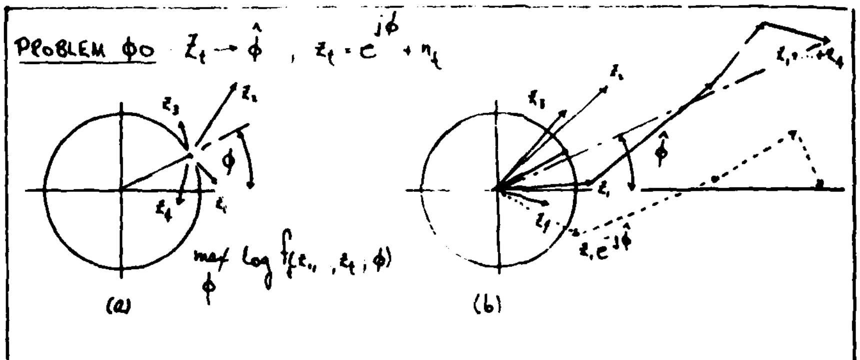

PROBLEM φ0 $Z_t \rightarrow \hat{\phi}$, $z_t = e^{j\phi} + n_t$
z_3
z_2
ϕ
z_4
z_1
$\max_{\phi} \log f(z_1,\ldots,z_t;\phi)$
(a)
z_2
z_3
$z_1+\cdots+z_4$
$\hat{\phi}$
z_1
z_t
$z_t e^{-j\hat{\phi}}$
(b)

Figure 1: *Constant Phase Estimation*

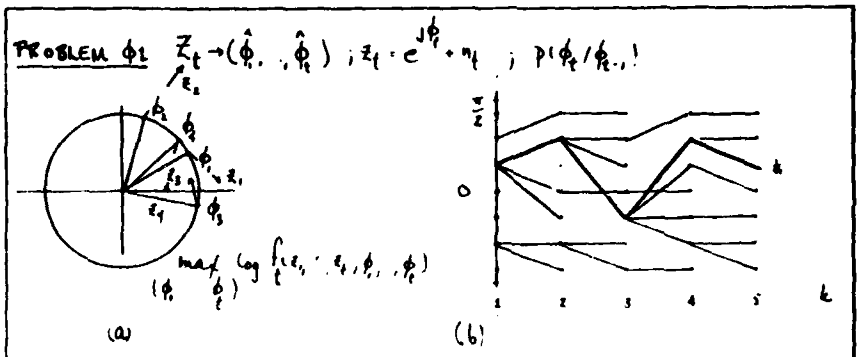

PROBLEM φ1 $Z_t \rightarrow (\hat{\phi}_1,\ldots,\hat{\phi}_t)$; $z_t = e^{j\phi_t} + n_t$; $p(\phi_t/\phi_{t-1})$
z_2
ϕ_2
ϕ_3
ϕ_1
z_1
z_3
z_4
ϕ_4
$\max_{(\phi_1,\ldots,\phi_t)} \log f(z_1,\ldots,z_t,\phi_1,\ldots,\phi_t)$
(a)
$\frac{\pi}{2}$
0
1
2
3
4
5
k
k
(b)

Figure 2: Random Walk Phase Estimation

Problem $\phi 0$: Constant Phase (Figure 1)

This problem involves mapping the data set Z_t into a phase estimate $\hat{\phi} \in [0,2\pi)$ when measurements are generated according to

$$z_k = \exp(j\phi) + n_k \quad , \quad k \in \{1,2,\ldots,t\}$$

It is a straightforward excerise in maximum likelihood (ML) theory to show

$$\hat{\phi} = \arg \sum_{k=1}^{t} z_k$$

As shown in Figure 1a this estimator maximizes the log-likelihood of $\{z_1,\ldots,z_t\}$:

$$\hat{\phi} = \arg\max \ell n \; f_t(z_1,\ldots,z_t;\phi)$$

Geometrically, the estimator is obtained by piecing measurements together, feather-to-tip, and measuring the angle to the resulting vector. This is illustrated in Figure 1b. The diagram in Figure 1c illustrates that if each measurement is rotated through an angle $\hat{\phi}$, and each rotated measurement added to the previous, the result is purely real.

Problem $\phi 1$: Random Walk Phase (Figure 2)

Here the problem is to map the data set Z_t into a phase sequence estimate $\{\hat{\phi}_1,\ldots,\hat{\phi}_t\} \in [0,2\pi)^t$ when measurements are generated according to

$$z_k = \exp(j\phi_k) + n_k \qquad k=1,2,\ldots,t$$

$$p(\phi_k/\phi_{k-1}) \text{ given}$$

As shown in Figure 2a, the MAP phase sequence maximizes the log-likelihood of $\{z_1,\ldots,z_t\}$:

$$(\hat{\phi}_1,\ldots,\hat{\phi}_t) = \arg \max_{\{\phi_k\}_1^t} \ell n \; f_t(z_1,\ldots,z_t,\phi_1,\ldots,\phi_t)$$

Here f is the joint density of the measurements and the phase sequence. The likelihood $\ell_t = \log f_t$ may be written

$$\ell_t = \ell_{t-1} - \frac{1}{2\sigma^2} z_t - e^{j\phi_t} + \log p(\phi_t/\phi_{t-1})$$

so if ω_t takes values in a discrete set (say $q2\pi/Q, q=0,1,\ldots,Q-1$) one can implement a dynamic programming algorithm on the lattice of Figure 2b to decode the MAP sequences. See [7] for details of the algorithm and a comparison of performance with other phase estimation algorithms.

1. FREQUENCY SEQUENCE ESTIMATION

Figures 3 and 4 depict two classical frequency estimation problems: constant frequency estimation and random walk frequency estimation.

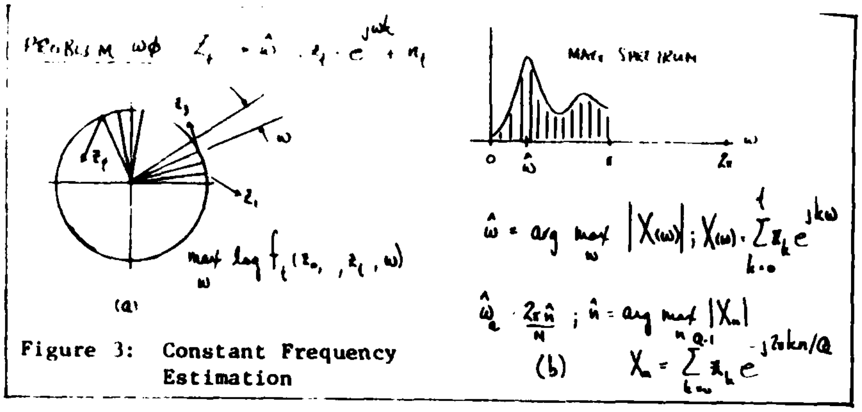

PROBLEM ωφ $Z_t \to \hat{\omega}$; $z_t = e^{j\omega k} + n_t$
z_1
z_t
z_0
ω
$\max_{\omega} \log f_t(z_0, \ldots, z_t, \omega)$
(a)
MAG. SPECTRUM
0
$\hat{\omega}$
π
2π
ω
$\hat{\omega} = \arg\max_{\omega} |X(\omega)|;\ X(\omega) = \sum_{k=0}^{t} z_k e^{jk\omega}$
$\hat{\omega}_Q = \frac{2\pi\hat{n}}{N};\ \hat{n} = \arg\max_{n} |X_n|$
(b)
$X_n = \sum_{k=0}^{Q-1} z_k e^{-j2\pi kn/Q}$

Figure 3: Constant Frequency Estimation

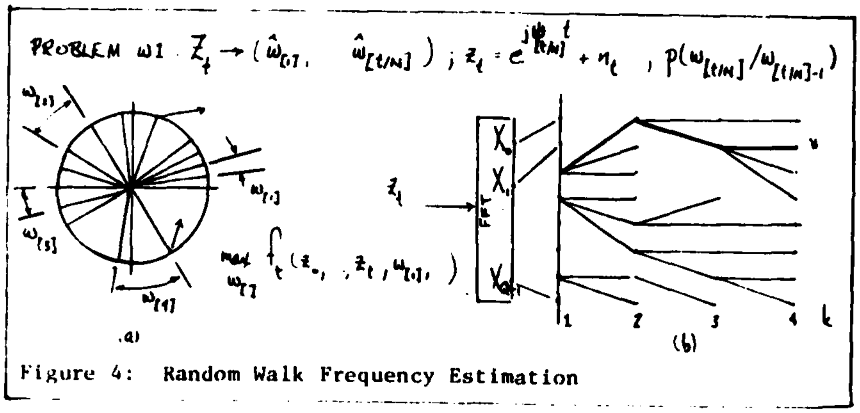

PROBLEM ω1 $Z_t \to (\hat{\omega}_{[1]}, \ldots \hat{\omega}_{[t/N]})$; $z_t = e^{j\psi_{[t/N]} t} + n_t$, $p(\omega_{[t/N]}/\omega_{[t/N]-1})$
$\omega_{[1]}$
$\omega_{[2]}$
$\omega_{[3]}$
$\omega_{[4]}$
$\max_{\omega_{[\,]}} f_t(z_0, \ldots, z_t, \omega_{[1]}, \ldots)$
(a)
z_t
X_0
X_1
X_{Q-1}
1
2
3
4
k
(b)

Figure 4: Random Walk Frequency Estimation

Problem 0: Constant Frequency Estimation (Figure 3)

This problem involves mapping the data set Z_t into a frequency estimate $\hat{\omega} \in [0,2\pi)$ when measurements are generated according to

$$z_k = \exp(j\omega k) + n_k \quad , \quad k \in \{0,1,\ldots,t\}$$

It is a straightforward exercise in ML theory to show

$$\hat{\omega} = \arg\max_{\omega} |X(\omega)|^2$$

$$X(\omega) = \sum_{k=0}^{t} z_k e^{jk\omega} \quad \text{(Fourier transform)}$$

As shown in Figure 3a this estimator maximizes the log-likelihood of $\{z_0,\ldots,z_t\}$:

$$\hat{\omega} = \arg\max_{\omega} f(z_0,\ldots,z_t;\omega)$$

An obvious approximation strategy, illustrated in Figure 3b, is to zero-pad the measurements and estimate ω as the DFT cell where a maximum occurs:

$$\hat{\omega} = \arg \max_{\omega_q = q2\pi/Q} \sum_{k=0}^{Q-1} z_k e^{-j2\pi qk/Q} \quad ,$$

$$z_k = 0 \quad , \quad k > t$$

Problem ω1: Random Walk Frequency Estimation (Figure 4)

Here the problem is to map the data set Z_t into a frequency sequence estimate $\{\hat{\omega}_{[0]},\ldots,\hat{\omega}_{[t/N]}\} \in [0,2\pi)^{[t/N]}$ when measurements are generated according to

$$z_k = \exp(j\omega_{[k/N]}k) + n_k \quad , \quad k=0,1,\ldots,t$$

$$p(\omega_{[k/N]}/\omega_{[k/N]-1}) \quad \text{given}$$

As shown in Figure 4a the MAP phase sequence maximizes the log-likelihood of $\{z_0,\ldots,z_t\}$.

$$(\hat{\omega}_{[0]},\ldots,\hat{\omega}_{[t/N]}) = \arg \max_{\{\omega_{[k/N]}\}_{k=0}^{t}} \ln f_t(z_0,\ldots,z_t;\omega_{[0]},\ldots,\omega_{[t/N]})$$

The likelihood $\ell_t = \log f_t$ may be written

$$\ell_t = \ell_{t-1} + \frac{1}{\sigma^2} \mathrm{Re}\, e^{-j\omega_{[t]}tN} X_t(\omega_{[t]}) + \ln p(\omega_{[t]}/\omega_{[t]-1})$$

Here X_t is DFT over the t^{th} data block of N samples. So if $\omega[t]$ takes values in a discrete set (say $q2\pi/Q, q=0,1,\ldots,Q-1$), one can implement a dynamic programming algorithm on the lattice of Figure 4b to decode the MAP sequence. See [8] for details.

4. CONCLUSIONS

The problems discussed here generalize. The basic idea is to select states and transition probabilities to characterize an underlying probabilistic structure, and then to assign characters (such as $e^{j\phi_k}$ or $e^{j\omega[k/N]k}$) to the states. The resulting sequence estimation algorithms are attractive because storage goes like Q (number of states) and computations are naturally parallel.

REFERENCES

[1] F. Lehan and R. Parks, "Optimum Demodulation," IRE National Convention Record, part 8, pp. 101-103 (1953).

[2] D. C. Youla, "The Use of Maximum Likelihood in Estimating Continuously Modulated Intelligence Which Has Been Corrupted by Noise," IRE Trans. Inform. Theory, IT-3, pp. 90-105 (March 1954).

[3] H. Cox, "Recursive Nonlinear Filtering," Proc. National Electronics Conference, vol. XXI, pp. 770-775 (1965).

[4] A. J. Viterbi, "Error Bounds for Convolutional Codes and An Asymptotically Optimum Decoding Algorithm," IEEE Trans. Inform. Theory, IT-15, pp. 260-269 (April 1969).

[5] C. R. Cahn, "Phase Tracking and Demodulation with Delay," IEEE Trans. Inform. Theory, IT-20, pp. 50-58 (Jan. 1974).

[6] G. D. Forney, "Maximum Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference," IEEE Trans. Inform. Theory, IT-18, pp. 363-378 (May 1972).

[7] L. L. Scharf, D. D. Cox, and C. J. Masreliez, "Modulo-2π Phase Sequence Estimation," IEEE Trans. Inform. Theory, in print.

[8] L. L. Scharf and H. Elliott, "A Random Sampler of Dynamic Programming Applications in Signal Processing and Control," Proc. Thirteenth Asilomar Conference on Cir., Syst., and Computers, pp. 7-13 (Nov. 5-7, 1979).

A DYNAMIC PROGRAMMING ALGORITHM FOR PHASE ESTIMATION AND DATA DECODING ON RANDOM PHASE CHANNELS

Odile Macchi and Louis L. Scharf

Reprinted from IEEE Transactions on Information Theory, Vol. IT–27, No. 5, September 1981
0018-9448/81/0900-0581$00.75 © 1981 IEEE

A Dynamic Programming Algorithm for Phase Estimation and Data Decoding on Random Phase Channels

ODILE MACCHI, MEMBER, IEEE, AND LOUIS L. SCHARF, SENIOR MEMBER, IEEE

***Abstract*—The problem of simultaneously estimating phase and decoding data symbols from baseband data is posed. The phase sequence is assumed to be a random sequence on the circle, and the symbols are assumed to be equally likely symbols transmitted over a perfectly equalized channel. A dynamic programming algorithm (Viterbi algorithm) is derived for decoding a maximum *a posteriori* (MAP) phase-symbol sequence on a finite dimensional phase-symbol trellis. A new and interesting principle of optimality for simultaneously estimating phase and decoding phase-amplitude coded symbols leads to an *efficient two-step decoding procedure* for decoding phase-symbol sequences. Simulation results for binary, 8-ary phase shift keyed (PSK), and 16-quadrature amplitude shift keyed (QASK) symbol sets transmitted over random walk and sinusoidal jitter channels are presented and compared with results one may obtain with a decision-directed algorithm or with the binary Viterbi algorithm introduced by Ungerboeck. When phase fluctuations are severe and when occasional large phase fluctuations exist, MAP phase-symbol sequence decoding on circles is superior to Ungerboeck's technique, which in turn is superior to decision-directed techniques.**

I. Introduction

PHASE FLUCTUATIONS can significantly increase the error probability for symbols transmitted over a channel that may or may not have been equalized. This is especially true for phase shift keyed (PSK) and quadrature amplitude shift keyed (QASK) symboling, in which case accurate phase discrimination is essential for symbol decoding. Even when the receiver contains a decision-directed phase-locked loop (DDPLL), performance loss in signal-to-noise ratio (SNR) with respect to a coherent decoding system can be in the range 5–10 dB. This fact is established in [1] for practical symbol sets and typical values of the phase variance parameter and symbol error probability.

On telephone lines, linear distortion and phase jitter dictate the use of a channel equalizer and some kind of phase estimator to achieve high rate, low error probability data transmission. A common approach to phase estimation and data decoding is to use a decision-directed algorithm in which a phase estimate is updated on the basis of old phase estimates and old symbol decisions. The DDPLL of [5] is a first-order digital phase-locked loop (PLL) in which the phase estimate is updated on the basis of a new measured phase and an old symbol decision. In the jitter equalizer (JE) of [3] and [4] a complex gain is updated according to a simple decision-directed stochastic ap-

Manuscript received December 26, 1979; revised November 17, 1980. This work was supported in part by the Centre National de la Recherche Scientifique; the Office of Naval Research, Statistics and Probability Branch, Arlington, VA; and the Army Research Office, Research Triangle Park, NC.

O. Macchi is with the CNRS Laboratoire des Signaux et Systèmes, Plateau du Moulon, 91190 Gif-sur-Yvette, France.

L. L. Scharf was with the CNRS Laboratoire des Signaux et Systèmes, Gif-sur-Yvette, France and the Electrical Engineering Department, Colorado State University, Fort Collins, CO. He is now with the Electrical Engineering Department, University of Rhode Island, Kingston.

0018-9448/81/0900-0581$00.75 ©1981 IEEE

proximation algorithm. The complex gain is used to scale and rotate the received signal, thereby correcting phase jitter and normalizing rapid fading variations. Although there is no explicit interest in phase estimation itself in the JE, it is possible to interpret the structure as an adaptive gain-phase correcting equalizer.

Both the DDPLL and the JE are very simple to implement, but apparently neither achieves optimality with respect to any statistical criterion for symbol (or data) decoding. Furthermore, neither the DDPLL nor the JE is optimum for estimating and/or correcting phase. Both are zero-lag phase estimators that cannot benefit from future signal samples. Therefore, an important question to be answered is whether or not symbol decoding can be improved using a better phase estimator. The answer, based on the results of [1] and this paper, is that significant improvements can be realized when the phase fluctuations are severe if one is willing to pay the price of an increased computational burden. In practice, cases of severe phase fluctuation can occur in high data rate PSK and QASK systems in which the angular distance between symbols is small.

In [1] Ungerboeck recognized the potential of maximum *a posteriori* (MAP) sequence estimation for jointly estimating phase and decoding data symbols. A path metric was derived and its role in a forward dynamic programming algorithm for obtaining MAP phase-symbol sequences was indicated. Because of the way phase was modeled in [1], the dynamic programming algorithm could not be solved directly. Ungerboeck approximated the phase sequence as a process that could make discrete binary jumps and then derived a dynamic programming algorithm for decoding likely paths around a developing most likely path. The result is a tree-search algorithm which may branch left or right but never go straight. He obtained performance results that were on the order of 3 dB superior in SNR to the DDPLL in a 16-QASK system, at interesting values of the phase variance parameter. We call the algorithm of [1] a discrete binary Viterbi algorithm (DBVA). The reader is referred also to [5] and [6] for discussions of other suboptimal, but computationally tractable, algorithms for simultaneously estimating phase and decoding data symbols.

In this paper we observe that baseband data is invariant to modulo-2π transformations on the phase sequence. This motivates us to wrap the phase around the circle, so to speak, and obtain folded probability models for transition probabilities on the circle. When the phase process is normal random walk on the circle, then the transition probabilities are described by a folded normal model. This model has also been used in [7] and [8]. It is then straightforward to pose a MAP sequence estimation problem for simultaneous phase and symbol sequence decoding as described in [8] and [9]. The basic idea is to discretize the phase space $[-\pi, \pi)$ to a finite dimensional grid and to use a dynamic programming algorithm (Viterbi algorithm) to keep track of surviving phase-symbol sequences that can ultimately approximate the desired MAP phase-symbol sequence. The MAP phase-symbol sequence itself is the entire sequence of past phases and symbols that is most likely, given an entire sequence of recorded observations. It is this use of "future" and "past" received signal samples that provides performance improvement over zero-lag estimators such as the DDPLL. Details of the algorithm are given in [8] and [9]. For PSK and QASK symbol sets an interesting principle of optimality leads to an efficient two-step decoding procedure. With this procedure, computational complexity is reduced by a factor greater than the number of admissible phase values per amplitude level. This amounts to a factor of four for the 16-point QASK diagram that has been recommended by CCITT for data transmission on telephone lines at 9600 bits/s. Finally, in order to make the computation and storage requirements tractable in the Viterbi algorithm, we use it in a fixed delay mode, as do other authors. By appealing to known results for fixed-lag smoothing of linearly observed data, we are able to intelligently choose the fixed delay. Without significant performance loss we decode phase-symbol pairs at a depth constant of $k_0 = 10$. This obviates the need for huge storage requirements for long sequences. With these modifications the Viterbi algorithm becomes a feasible, albeit sophisticated, decoding procedure.

Simulation results for the proposed Viterbi algorithm (VA) are presented for several symbol sets consisting of two, eight, or 16 symbols. Several types of phase jitter are investigated such as Gaussian and non-Gaussian random walk and sinusoidal phase jitter. The resulting error probabilities are compared with those of the simpler decision-directed algorithms (JE and DDPLL) and with those of the DBVA. As expected, performance of the VA is always superior to that of the other systems. On the other hand, the increase in computational burden is substantial, and the improvement in performance is not always great enough to warrent the use of the VA. In our concluding remarks we discuss situations in which one might reasonably use the VA or the DBVA rather than a simpler decision-directed algorithm such as the JE or the DDPLL.

Remarks on Notation:

Throughout this paper $\perp\!\!\!\perp$ denotes statistical independence. The notation $\{\phi_k\}_1^K$ will mean the set $\{\phi_k, k = 1, 2, \cdots, K\}$. When the indexes 1 and K are missing (e.g., $\{\phi_k\}$), it is understood that K is infinite. The symbol N^+ denotes the positive integers. The notation $x : N_x(\mu, \sigma^2)$ means the random variable x is normally distributed with mean μ and variance σ^2; $N_x(\mu, \sigma^2)$ will also be used to denote the function $(2\pi\sigma^2)^{-1/2} \exp\{-(x-\mu)^2/2\sigma^2\}$. When x is complex, $x : N_x(\mu, \sigma^2)$ means x is complex with density $N_x(\mu, \sigma^2) = (2\pi\sigma^2)^{-1} \exp\{-|x-\mu|^2/2\sigma^2\}$. By $f(x/y)$ we mean the conditional probability density of the random variable x, given the random variable y. Thus $f(x/y)$ is generally a different function than $f(w/z)$, even though we use no explicit subscripting such as $f_{w/z}(\cdot/\cdot)$ to indicate so. We make no notational distinction between a random variable and its realizations, relying instead on context to make the meaning clear. A density function for a random variable, evaluated at a particular realization of the random variable is termed a likelihood function.

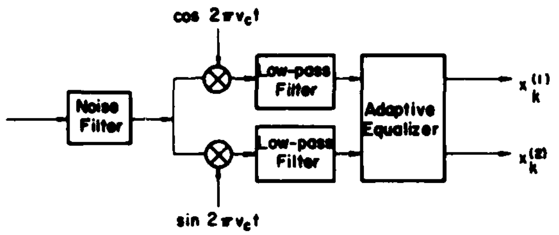


Fig. 1. Typical signal receiver for data transmission.

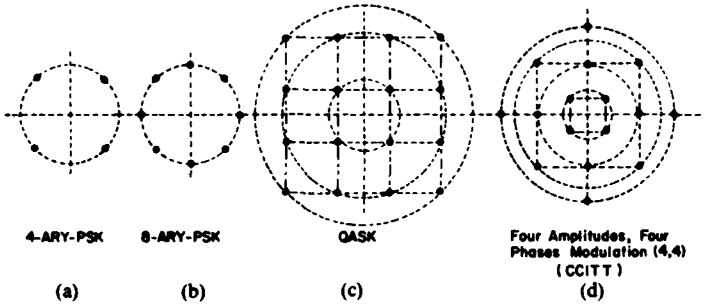


Fig. 2. Symbol diagrams for PSK and QASK modulation schemes.

"Hatted" variables such as $\hat{\phi}_k$ refer always to MAP estimates that maximize an *a posteriori* density. Finally, it is convenient to define the function

$$g_M(x) = M^{-1} \sum_{m=1}^{M} \sum_{l=-\infty}^{\infty} h[x - l2\pi - (m-1)2\pi/M] \tag{1}$$

where $h(\cdot)$ is a probability density. The function $g_M(\cdot)$ plays an important role in our discussion of phase-symbol decoding on QASK symbol sets.

II. Signal and Phase Models

Assume complex data symbols $\{a_k\}$ are phase or phase-amplitude modulated onto a carrier and transmitted over a channel with linear distortion and phase jitter. The received signal—call it $y(t)$—is typically processed as illustrated in Fig. 1. The signal $y(t)$ is passed through a bandpass noise filter and demodulated with two quadrature waveforms. The resulting complex baseband signal $x_1(t) + jx_2(t)$ is equalized with a complex adaptive equalizer in order to reduce the intersymbol interference due to linear distortion in the channel. The equalized signal is a sequence of samples at symbol rate $1/\Delta$ (Δ is the interval between successive data symbols). The output of the equalizer is a complex sequence $x_k = x_k^{(1)} + jx_k^{(2)}$ which is a noisy, phase-distorted, version of the original transmitted sequence. Thus we write

$$x_k = a_k e^{j\phi_k} + n_k, \qquad k \in N^+. \tag{2}$$

Here, $\{a_k\}$ is the complex symbol sequence, typically encoded according to one of the diagrams illustrated in Fig. 2. The sequence $\{\phi_k\}$ represents phase fluctuations (jitter and frequency drift) in the channel. The two real components $n_k^{(1)}$ and $n_k^{(2)}$ of the complex noise sequence $n_k = n_k^{(1)} + jn_k^{(2)}$ are the noise variables in the respective baseband quadrature equalized channels. The variables $n_k^{(1)}$ and $n_k^{(2)}$ can be shown to be independent when the carrier frequency is in the middle of the input noise filter bandwidth and the additive channel noise is white. If the equalizer is perfect, then n_k is the usual Gaussian, additive noise with zero-mean. If the equalizer is not perfect, then n_k contains a residual of the intersymbol interferences, and is not Gaussian; nor are successive variables $n_k^{(1)}, n_{k+1}^{(1)}, \cdots$, independent. However, for a reasonably good equalizer, we may assume that $\{n_k\}$ is a sequence of independent identically distributed (i.i.d.) complex Gaussian variables. Strictly speaking, this assumption is valid only at the input to the equalizer when the baseband equivalent of the input noise filter and low-pass demodulator is the so-called sampled whitened matched filter of [10]. In practice, the assumption of Gaussianity is more realistic than the assumption of independence for the sequence $\{n_k\}$. Assuming that the equalizer of Fig. 1 is perfect, we model the noise sequence $\{n_k\}$ as follows:

$$\begin{aligned} & n_k = n_k^{(1)} + jn_k^{(2)}, \qquad k \in N^+ \\ & n_k^{(1)} \perp\!\!\!\perp n_l^{(2)}, \qquad \forall\,(k,l) \\ & n_k^{(1)} \perp\!\!\!\perp n_l^{(1)}, \qquad k \neq l, \qquad n_k^{(2)} \perp\!\!\!\perp n_l^{(2)}, \qquad k \neq l \\ & n_k^{(1)} : N_{n_k}(0, \sigma_n^2); \qquad n_k^{(2)} : N_{n_k}(0, \sigma_n^2). \end{aligned} \tag{3}$$

Here $2\sigma_n^2$ is the variance of the complex noise variable n_k, and σ_n^2 is the variance of each real component.

Consider now the phase distortion $\{\phi_k\}$. The term generally reflects two effects, one long-term and the other short-term. In modern high speed data modems no carrier or pilot tone is transmitted for locking the local oscillator at the receiver. Thus long-term large-range linear phase variations result from frequency drift in the channel which cannot be eliminated. In addition, nonlinear intermodulation with local power supplies gives rise to short-term

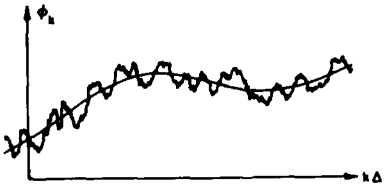


Fig. 3. Typical phase fluctuations: phase jitter and frequency drift.

small-range phase variations. The variations exhibit energetic harmonic content at the harmonics of the fundamental power supply frequency. Hence a realistic model for $\{\phi_k\}$ is

$$\phi_k = (\phi_0 + 2\pi Bk) + \sum_{l=1}^{p} A_l \sin(2\pi\nu_l k\Delta + \rho_l), \qquad k \in N^+ \tag{4}$$

where $\nu_l = l \cdot 50$ Hz or $\nu_l = l \cdot 60$ Hz, depending on the place of use. A typical phase process is depicted in Fig. 3. The first term in parentheses in (4) is the so-called frequency drift term and the summation term is the phase jitter. In practice, the constants ϕ_0, B, $\{A_l, \nu_l, \rho_l\}_{l=1}^{P}$ vary with time $k\Delta$ but at an extremely slow rate.

The spectrum of the phase jitter, i.e., the behavior of A_l versus ν_l, has been investigated experimentally in [14]. The spectrum is roughly fitted by a $1/\nu^2$ curve. A phenomenological model for phase having a $1/\nu^2$ spectrum (like that of phase jitter at high frequencies) is the Wiener-Levy continuous time process,

$$\frac{d\phi(t)}{dt} = w(t), \qquad t \geq 0, \tag{5}$$

where $\{w(t)\}$ is a white noise process. The discrete time analog is the independent increments sequence

$$\phi_k = \phi_{k-1} + w_k, \qquad k \in N^+ \tag{6}$$

where $\{w_k\}$ is a sequence of i.i.d. random variables with even probability density $h(w)$.[1] When $w_k : N_w(0, \sigma_w^2)$, then $\{\phi_k\}$ is the so-called normal random walk.

For short-term fluctuations, the model captures, with appropriate selection of $h(w)$, the correlated evolution of phase. The main virture of the independent increments model is that it forms a convenient basis from which to derive optimum estimator structures which may then be evaluated against more realistic phase sequences.

Since the measurement model of (2) is invariant to modulo-2π translates of ϕ_k, we may represent phase as if it were a random sequence on the unit circle C or equivalently on the interval $[-\pi, \pi)$. Call $\bar{\phi}_k$ this representation of ϕ_k. Note $\bar{\phi}_{k+1}$ may be written

$$\bar{\phi}_{k+1} = \bar{\phi}_k + \bar{w}_k \tag{7}$$

where the plus sign denotes modulo-2π addition of real variables or equivalently rotation with positive (counterclockwise) sense on C. The variable $\bar{w}_k$ is a modulo-2π version of w_k.

The conditional density of $\bar{\phi}_{k+1} \triangleq \bar{\phi}_k + w_k$, given $\bar{\phi}_k$, is $h(\bar{\phi}_{k+1} - \bar{\phi}_k)$. Since $\bar{\phi}_{k+1}$ is a modulo-2π version of ϕ_{k+1}, we may reflect all of the conditional probability mass into C to obtain the transition (or conditional) probability density

$$f(\bar{\phi}_{k+1}/\bar{\phi}_k) = \sum_{l=-\infty}^{\infty} h(\bar{\phi}_{k+1} - \bar{\phi}_k - l2\pi)$$
$$= g_1(\bar{\phi}_{k+1} - \bar{\phi}_k) \tag{8}$$

where g_1 is the function defined in (1). Hereafter, $g_1(\cdot)$ is called the folded density of the phase increments. Usually, the phase increment is small and its distribution $h(\cdot)$ is very narrow with respect to 2π. Therefore, in the sum of (8) only one term is relevant and $f(\bar{\phi}_{k+1}/\bar{\phi}_k) \doteq h(\bar{\phi}_{k+1} - \bar{\phi}_k)$. In the normal case, this implies $\sigma_w \ll 2\pi$, where σ_w^2 is the variance of w_k. As it is cumbersome to carry around the overbar notation $\bar{\phi}_{k+1} - \bar{\phi}_k$, we drop it with the caution that from here on ϕ_k is defined on C unless otherwise stated.

In the normal case [7], [8], the density $g_1(\phi_{k+1} - \phi_k)$ may be written

$$g_1(\phi_{k+1} - \phi_k) = \sum_{l=-\infty}^{\infty} N_{\phi_{k+1}}(\phi_k + l2\pi, \sigma_w^2). \tag{9}$$

This case and the Cauchy case (in which the distribution tails are much heavier than the normal tails) are studied in the Appendix. It is shown that $g_1(x)$ achieves its maximum at $x = 0$ and that it is monotonically decreasing on $0 \leq x \leq \pi$.

The sequence $\{\phi_k\}_1^K$ is Markov. Therefore, we may write for the joint density of the K phases $\{\phi_k\}_1^K$

$$f(\{\phi_k\}_1^K) = \prod_{k=0}^{K-1} f(\phi_{k+1}/\phi_k)$$
$$f(\{\phi_1/\phi_0\}) \triangleq f(\phi_1) : \text{the marginal density of } \phi_1. \tag{10}$$

Usually, ϕ_1 is uniformly distributed on C because phase acquisition starts at $k = 1$ with no prior information about its value. By the independence of the n_k in (2), it follows that the conditional density of the measurement sequence $\{x_k\}_1^K$, given the phase and data sequences $\{\phi_k\}_1^K$, $\{a_k\}_1^K$, is

$$f(\{x_k\}_1^K / \{\phi_k\}_1^K, \{a_k\}_1^K) = \prod_{k=1}^{K} N_{x_k}(a_k e^{j\phi_k}, \sigma_n^2). \tag{11}$$

Equations (8)–(11) form the basis for the derivation of a MAP sequence estimator. The key element is that $\{\phi_k\}$ is a Markov sequence with a bounded range space $[-\pi, \pi)$. Discretization of this bounded interval leads to a finite-state model from which a finite dimensional dynamic programming algorithm can be derived.

III. Decision-Directed Algorithms

The usual way of dealing with phase fluctuations is to design a phase estimator and use the estimated phase, call it $\hat{\phi}_k$, to rotate the received signal as follows:

$$y_k = x_k e^{-j\hat{\phi}_k}, \qquad k \in N^+. \tag{12}$$

The phase corrected signal y_k is then fed to a decision device which, in turn, delivers the symbol estimate $\hat{a}_k$.

[1] That is, $h(w) = h(-w)$.

Typically, the phase estimate $\hat{\phi}_k$ is functionally dependent on the old measurements $\{\cdots, x_{k-2}, x_{k-1}\}$ and the past symbol estimates $\{\cdots, \hat{a}_{k-2}, \hat{a}_{k-1}\}$. If a carrier or pilot tone is transmitted as in typical single sideband (SSB) systems, then $\hat{\phi}_k$ is obtained from a simple phase-locked loop (PLL). In suppressed carrier systems such as PSK or QASK systems, the PLL is decision-directed. That is, $\hat{\phi}_k$ is updated on the basis of $\hat{a}_{k-1}$. For instance in [5]

$$\begin{aligned} \hat{\phi}_{k+1} &= \hat{\phi}_k + \mu \,\mathrm{Im}\left[x_k \hat{a}_k^* e^{-j\hat{\phi}_k}\right] \\ &\doteq \hat{\phi}_k + \mu_k \sin\left(\arg x_k - \arg \hat{a}_k - \hat{\phi}_k\right), \\ \mu_k &= \mu |x_k|^2 \qquad (13) \end{aligned}$$

where the asterisk denotes complex conjugate and μ is a constant that depends on the SNR. The estimator of (13) is called a DDPLL.

In the jitter equalizer (JE) of [3] and [4], x_k is rotated *and* scaled as follows:

$$\begin{aligned} y_k &= x_k G_k \\ G_k &= G_{k-1} + \mu(\hat{a}_{k-1} - y_{k-1}) x_{k-1}^*. \qquad (14) \end{aligned}$$

The complex gain G_k is the single complex coefficient of a one-coefficient rapidly adaptive equalizer. We may think of $G_k/|G_k|$ as the phase correction $e^{-j\hat{\phi}_k}$, and $|G_k|$ as a gain correction $\hat{c}_k$. Thus, although there is no explicit formulation of a phase-gain estimation problem in [3] and [4], the net effect of the JE is to correct phase *and* normalize rapid fading variations. As explained in [4], when phase fluctuations are large, the JE performance may be improved by setting a constraint on G_k that keeps its value inside a given domain including the complex point (0, 1).

Geometrical Comments

The combined effects of random phase fluctuations and additive noise may be illustrated as in Fig. 4(a). The transmitted symbol $a_k = a^{(0)}$ (say) is rotated by the random phase angle ϕ_k to give $a_k e^{j\phi_k}$. To this is added the complex noise sample n_k to give the measurement x_k defined in (2). For the case illustrated, the resultant measurement is closer to symbol $a^{(1)}$ than to $a^{(0)}$ and consequently, with no phase or phase-gain correction, a decoding error would be made. To emphasize the combined effects of phase fluctuation and additive noise, we have illustrated a case for which either phase jitter or additive noise alone would cause no error. See [11] for a probabilistic discussion of this issue. Fig. 4(b) is an illustration of how a DDPLL works. The angle ψ_k is the noisy measured phase ($\arg x_k$) minus the sum of the phase of the decoded symbol and the previously estimated phase ($\arg \hat{a}_k + \hat{\phi}_k$). A given amount μ_k of this angle is added to $\hat{\phi}_k$ as a correction to get the new phase estimate $\hat{\phi}_{k+1} = \hat{\phi}_k + \mu_k \psi_k$. Note that only phase is corrected. In the JE both phase and gain are corrected, offering potential for improved performance. This potential is particularly important in QASK symbol sets where amplitude errors in x_k can result in decoding errors.

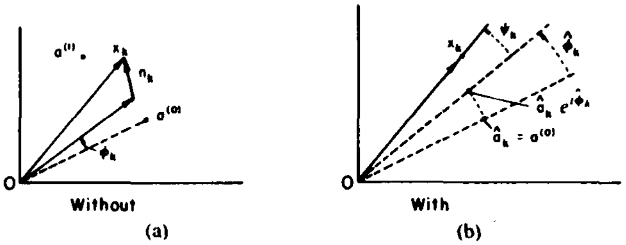


Fig. 4. Geometry of phase jitter and additive noise with and without phase correction of DDPLL. (a) Without. (b) With.

IV. MAP Phase and Symbol Sequence Decoding with the Viterbi Algorithm

The basic idea behind MAP sequence decoding is to find a sequence of phase-symbol pairs $\{\phi_k, a_k\}_1^K$ that, based on the observation sequence $\{x_k\}_1^K$, appears most likely. The application of this idea to data communication was first proposed in [1] and refined in [9]. The most likely sequence, call it $\{\hat{\phi}_k, \hat{a}_k\}$, is the sequence that maximizes the natural logarithm (or any other monotone function) of the *a posteriori* density of $\{\phi_k, a_k\}_1^K$, given the sequence of observations $\{x_k\}_1^K$. Thus we pose the maximization problem:

$$\max_{\{\phi_k\}_1^K, \{a_k\}_1^K} \ln f\left(\{\phi_k\}_1^K, \{a_k\}_1^K / \{x_k\}_1^K\right). \qquad (15)$$

This is equivalent to maximizing the natural logarithm of the likelihood function $f(\{x_k\}_1^K, \{\phi_k\}_1^K, \{a_k\}_1^K)$, obtained by evaluating the joint density function for $\{x_k\}_1^K$, $\{\phi_k\}_1^K$, and $\{a_k\}_1^K$, at the observed values of $\{x_k\}_1^K$. Using the results of (10) and (11) we may write

$$\begin{aligned} &f\left(\{x_k\}_1^K, \{\phi_k\}_1^K, \{a_k\}_1^K\right) \\ &\quad = \left[\prod_{k=1}^{K} N_{x_k}\left(a_k e^{j\phi_k}, \sigma_n^2\right) f(\phi_k/\phi_{k-1})\right] f\left(\{a_k\}_1^K\right). \qquad (16) \end{aligned}$$

Assuming the $\{a_k\}_1^K$ to be a sequence of independent, equally likely symbols, using (8), and neglecting irrelevant constants, we may write the maximization problem as

$$\max_{\{\phi_k\}_1^K, \{a_k\}_1^K} \Gamma_K$$

$$\begin{aligned} \Gamma_K &= -\frac{1}{2\sigma_n^2} \sum_{k=1}^{K} |x_k - a_k e^{j\phi_k}|^2 \\ &\quad + \sum_{k=2}^{K} \ln g_1(\phi_k - \phi_{k-1}) + \ln f(\phi_1). \qquad (17) \end{aligned}$$

Note that Γ_k satisfies the recursion

$$\begin{aligned} \Gamma_k &= \Gamma_{k-1} + p_k, \qquad k = 2, 3, \cdots \\ p_k &= -\frac{1}{2\sigma_n^2} |x_k - a_k e^{j\phi_k}|^2 + \ln g_1(\phi_k - \phi_{k-1}), \\ &\qquad\qquad k = 2, 3, \cdots \\ \Gamma_1 &= -\frac{1}{2\sigma_n^2} |x_1 - a_1 e^{j\phi_1}|^2 + \ln f(\phi_1) \qquad (18) \end{aligned}$$

where p_k is the so-called path-metric. For convenience, let us make explicit in Γ_K the last phase and symbol: $\Gamma_K(\phi_K, a_K)$. The other arguments $\{\phi_k\}_1^{K-1}$, $\{a_k\}_1^{K-1}$, remain implicit. Then, from (18)

$$\Gamma_K(\phi_K, a_K) = \Gamma_{K-1}(\phi_{K-1}, a_{K-1}) + p_K(x_K, a_K, \phi_K, \phi_{K-1}). \quad (19)$$

Thus the maximizing sequence—call it $(\{\hat{\phi}_k\}_1^K, \{\hat{a}_k\}_1^K)$—passing through $(\hat{\phi}_{K-1}, \hat{a}_{K-1})$ on its way to $(\hat{\phi}_K, \hat{a}_K)$, must arrive at $(\hat{\phi}_{K-1}, \hat{a}_{K-1})$ along a route $(\{\hat{\phi}_k\}_1^{K-2}, \{\hat{a}_k\}_1^{K-2})$ that maximizes $\Gamma_{K-1}(\hat{\phi}_{K-1}, \hat{a}_{K-1})$. It is this observation which forms the basis of forward dynamic programming. In the actual implementation of a dynamic programming algorithm, one must discretize the phase space C to a finite dimensional grid of phase values $\Xi = \{\xi_n\}_{n=1}^m$. The function $\ln g_1(\phi_k - \phi_{k-1})$ is then defined on the two-dimensional grid $\Xi \times \Xi$. However, as discussed in [8] and [9], the resulting $m \times m$ matrix of conditional probabilities has Toeplitz symmetry which means only an m vector of conditional probabilities must be computed and stored.

The Viterbi algorithm for simultaneous phase and symbol decoding consists simply of an algorithm which determines survivor phase-symbol sequences terminating at each possible phase-symbol pair. One of these surviving sequences is ultimately decoded as the approximate MAP phase-symbol sequence. The complexity c of the algorithm lies mainly in the evaluation of the mM possible values of $|x_k - a_k e^{j\phi_k}|^2$, for each new measurement x_k. Here M is the symboling alphabet size, and m is the number of discrete phase values. For each calculation of $|x_k - a_k e^{j\phi_k}|^2$ there are six real multiplies. Compared to this multiplication load of $6mM$ per sample, the determination and addition of the m possible values of $\ln g_1(\phi_k - \phi_{k-1})$ that appear in (18) is negligible. The determination of $|x_k - a_k e^{j\phi_k}|^2$ would likely be computed in a pipelined parallel architecture, while the terms $\ln g_1(\cdot)$ would be read by appropriately addressing read only memory (ROM). When short-term phase fluctuations have small amplitude (σ_w small) so that m must be large for accurate phase tracking, the complexity increases. For example, with $M = 8$ and $m = 48$, $c \sim 384$, indicating on the order of 2×10^3 computations at each k-step.

As we show in the next section, the complexity of the Viterbi algorithm can be dramatically reduced by making a change of variable and tracking a total phase variable that is the sum of ϕ_k and the symbol phase, $\arg a_k$. Also, of course, for PSK symbol sets only one symbol amplitude is admissible, and admissible symbol phases may be chosen to fall on one of the discrete phase values. Thus for PSK symbol sets the complexity is simply m, and the number of path metric computations is on the order of 300 for $m = 48$. Even this figure may be reduced by using one of a variety of so-called M algorithms in which all surviving phase-symbol pairs are saved, but only a handful of candidate originator pairs are considered for each survivor [16]–[18].

V. A Principle of Optimality for Phase-Amplitude Coded Symbols and an Efficient Two-Step Decoding Procedure

In order to simplify matters and to illustrate the key ideas, let us consider PSK symbols of the form

$$a_k = e^{j\theta_k} \quad (20)$$

with $\{\theta_k\}$ drawn independently from an M-ary equiprobable alphabet $\Theta = \{(l-1)2\pi/M\}_{l=1}^M$. Write the measurement model of (2) as

$$x_k = e^{j\psi_k} + n_k \quad (21)$$

where the total phase ψ_k is represented as

$$\psi_k = \phi_k + \theta_k$$
$$\theta_k = \sum_{l=1}^k \Delta\theta_l, \quad \Delta\theta_k = \theta_k - \theta_{k-1}, \quad \Delta\theta_1 = \theta_1. \quad (22)$$

It is clear that $\hat{\theta}_k = \Sigma_{l=1}^k \Delta\hat{\theta}_l$ and $\hat{\phi}_k = \hat{\psi}_k - \hat{\theta}_k$. Thus we may replace the MAP sequence estimation problem posed in (15) by the problem

$$\max_{\{\psi_k\}_1^K, \{\Delta\theta_k\}_1^K} f\left(\{x_k\}_1^K, \{\psi_k\}_1^K, \{\Delta\theta_k\}_1^K\right). \quad (23)$$

The joint density $f^K \triangleq f(\cdot, \cdot, \cdot)$ in (23) may be written

$$f^K = \prod_{k=1}^K N_{x_k}(e^{j\psi_k}, \sigma_n^2) f\left(\psi_k, \Delta\theta_k / \{\psi_j\}_1^{k-1}, \{\Delta\theta_j\}_1^{k-1}\right) \quad (24)$$

where for $k = 1$, $f(\psi_1, \Delta\theta_1 / \cdot, \cdot)$ is simply the marginal density $f(\psi_1, \Delta\theta_1)$. The conditional density on the right-hand side of (24) is easily evaluated with Bayes' rule:

$$f\left(\psi_k, \Delta\theta_k / \{\psi_j\}_1^{k-1}, \{\Delta\theta_j\}_1^{k-1}\right) = f\left(\psi_k / \{\psi_j\}_1^{k-1}, \{\Delta\theta_j\}_1^k\right) \cdot f\left(\Delta\theta_k / \{\psi_j\}_1^{k-1}, \{\Delta\theta_j\}_1^{k-1}\right). \quad (25)$$

Now $\Delta\theta_k$ is independent of the previous data, additive noise and phase fluctuations. Thus

$$f\left(\Delta\theta_k / \{\psi_j\}_1^{k-1}, \{\Delta\theta_j\}_1^{k-1}\right) = \frac{1}{M}. \quad (26)$$

Moreover, if we rewrite ψ_k as

$$\psi_k = \phi_{k-1} + w_k + \theta_{k-1} + \theta_k - \theta_{k-1}$$
$$= \psi_{k-1} + \Delta\theta_k + w_k, \quad (27)$$

we see immediately that

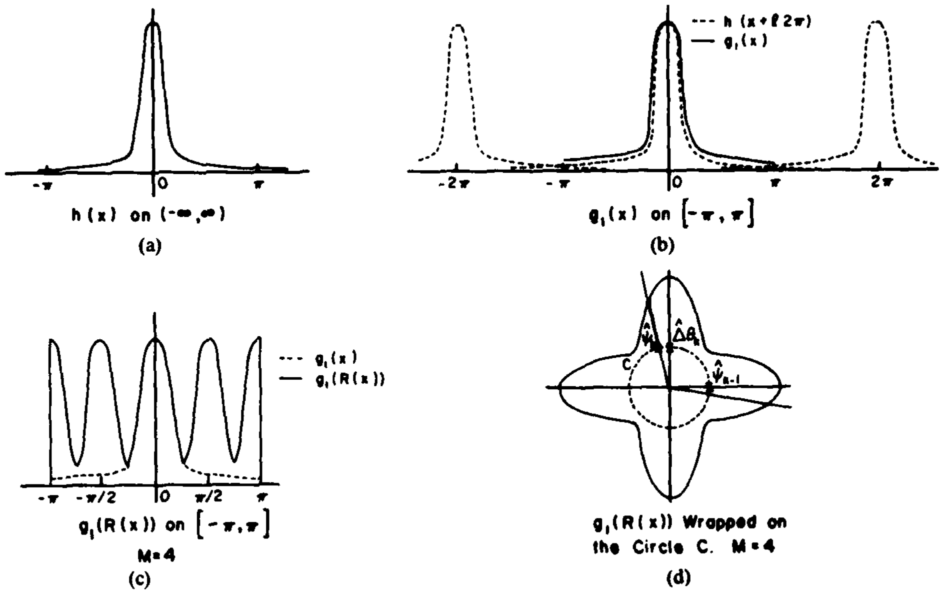
$$f\left(\psi_k / \{\psi_j\}_1^{k-1}, \{\Delta\theta_j\}_1^k\right) = g_1(\psi_k - \psi_{k-1} - \Delta\theta_k). \quad (28)$$

Recall ψ_k is defined on the circle C. Therefore, for clarity we might think of ψ_k as a random variable $\psi_{k-1} + \Delta\theta_k + w_k$, whose density is folded in $[-\pi, \pi)$. Putting (24)–(28) together, we have for the joint density f^K

$$f^K = \prod_{k=1}^K N_{x_k}(e^{j\psi_k}, \sigma_n^2) \frac{1}{M} g_1(\psi_k - \psi_{k-1} - \Delta\theta_k)$$
$$\Delta\theta_1 \triangleq \theta_1, \quad \psi_0 \triangleq 0. \quad (29)$$

Principle of Optimality

Call $\{\hat{\psi}_k\}_1^K, \{\Delta\hat{\theta}_k\}_1^K$ the MAP sequences that maximize f^K; $\{\Delta\theta_k\}_1^K$ enters only in the $g_1(\cdot)$ term on the right-hand side of (29). Now let us suppose (as is usual) that $g_1(w)$, which is even, is also unimodal with a peak at $w = 0$. This single-mode assumption for $g_1(\cdot)$ is valid in particular when the phase increment w_k in the Markov process (6) has a Gaussian or Cauchy distribution $h(w)$ (see the Appen-



h(x) on (-∞,∞)
(a)

g₁(x) on [-π,π]
(b)

g₁(R(x)) on [-π,π]
M=4
(c)

g₁(R(x)) Wrapped on the Circle C. M=4
(d)

Fig. 5. Density functions of phase increment before and after folding.

dix). It follows that f^K is maximized by choosing

$$\Delta\hat{\theta}_k = [\hat{\psi}_k - \hat{\psi}_{k-1}] \tag{30}$$

where $[x]$ denotes the closest value of $(l - 1)2\pi/M$ to x. By substitution of the constraint (30) into (29) and defining the "rest" function $R(x)$ on the circle C by

$$R(x) = x - [x], \tag{31}$$

we find that one must maximize

$$\hat{f}^K = \prod_{k=1}^{K} N_{x_k}(e^{j\psi_k}, \sigma_n^2)\frac{1}{M}g_1(R(\psi_k - \psi_{k-1})). \tag{32}$$

The maximization of $\hat{f}^K$ with respect of $\{\psi_k\}_1^K$ is formally equivalent to maximizing the joint density $f(\{x_k\}_1^K, \{\psi_k\}_1^K)$ when the total phase ψ_k follows a Markov-model similar to (6):

$$\psi_k = \psi_{k-1} + u_k. \tag{33}$$

Here the independent increments u_k have "probability density," folded on the circle C,

$$f(u) = \frac{1}{M}g_1(R(u)). \tag{34}$$

This interpretation is purely formal since $f(u)$ is not generally a probability density. However, when

$$g_1(u) = 0, \qquad |u| \geq \frac{\pi}{M} \tag{35}$$

then $f(u)$ is a probability density because in that case

$$\frac{1}{M}g_1(R(u)) = g_M(u). \tag{36}$$

Thus (34) can be interpreted as an approximate density when the peak of $g(u)$ is narrower than the minimum phase distance between the symbols. This condition is always satisfied in communications applications; otherwise, phase distortion is so large that data transmission is not possible. Thus we have a pure phase-tracking problem as in [8] and [9], and we may proceed accordingly. Taking the natural logarithm of $\hat{f}^K$, we have the maximization problem

$$\max_{\{\psi_k\}_1^K} \Gamma_K',$$
$$\Gamma_k' = \Gamma_{k-1}' + p_k';$$
$$\Gamma_1' = -\frac{1}{2\sigma_n^2}|x_1 - e^{j\psi_1}|^2 + \ln g_1(R(\psi_1))$$
$$p_k' = -\frac{1}{2\sigma_n^2}|x_k - e^{j\psi_k}|^2 + \ln g_1[R(\psi_k - \psi_{k-1})] \tag{37}$$

which is solved by the dynamic programming algorithm discussed in Section IV. The complexity c' of this algorithm lies essentially in the evaluation of the m possible values of $|x_k - e^{j\psi_k}|^2$ for each new data value x_k. The m different values of $\ln g_1[R(\cdot)]$ will be precomputed and stored in ROM. For each computation of $|x_k - e^{j\psi_k}|^2$ there are two multiplies, so complexity is simply proportional to m. This represents a reduction in complexity greater than M for M-ary PSK.

Usually, the phase is differentially modulated rather than directly modulated, and therefore the relevant symbol is $\Delta\hat{\theta}_k$ itself (see (30)). For the purpose of data transmission there is no need to reconstruct the absolute data phase $\hat{\theta}_k = \Sigma_{l=1}^k \Delta\hat{\theta}_l$. This reconstruction has, however, been carried out in the simulations in order to recover the estimates $\hat{\phi}_k = \hat{\psi}_k - \hat{\theta}_k$ of the phase fluctuations and to get the approximate variance of the phase estimates

$$\hat{\sigma}_\phi^2 = \frac{1}{K}\sum_{k=1}^{K} |\phi_k - \hat{\phi}_k|^2. \tag{39}$$

Density Functions and Geometrical Comments

The entire development of this section has a nice geometric interpretation which we illustrate in Fig. 5. In Fig. 5(a) the basic phase noise density $h(x)$ is illustrated on

$(-\infty, \infty)$. Fig. 5(b) is the folded version $g_1(x)$ of $h(x)$ to account for the wrapping on the unit circle C. Fig. 5(c) is the function $g_1[R(x)]$ that arises in our discussion of the principle of optimality, sketched in the case of 4-ary phase modulation. Fig. 5(d) shows $g_1[R(x)]$ wrapped around the circle C. Since $g_1(x)$ is very narrow, $g_1[R(x)]$ is approximately the repeated copy of $g_1(x)$ at all possible values of data phase. With $x = \hat{\psi}_k - \hat{\psi}_{k-1}$, Fig. 5(d) illustrates the choice of $\Delta\hat{\theta}_k$ nearest $\hat{\psi}_k - \hat{\psi}_{k-1}$ ($\Delta\hat{\theta}_k = \pi/2$ is the best choice here), and the resulting value of $g_1[R(\hat{\psi}_k - \hat{\psi}_{k-1})]$ is shown by the heavy segment on the axis $\hat{\psi}_k$, terminated by the heavy dot.

We now extend this principle of optimality to phase-amplitude encoded symbols. Assume the independent, equally probable data symbols are complex symbols of the form

$$a_k = A_k e^{j\theta_k} \tag{40}$$

with the A_k positive real numbers drawn independently from the alphabet $A = (\alpha_1, \alpha_2, \cdots, \alpha_L)$. Denote by $p(A_k)$ the probability mass function for the random variable A_k. Assume the θ_k are drawn from the alphabet $B = (\beta_1, \beta_2, \cdots, \beta_{M_1})$. Denote the conditional probability mass function of θ_k, given A_k, by $p(\theta_k/A_k)$. For the (4, 4) diagram of Fig. 2(d),

$$A = \left(\sqrt{2}\,\alpha_1, 3\alpha_1, 3\sqrt{2}\,\alpha_1, 5\alpha_1\right);$$

$$B = \{b_i\}_{i=1}^{8}, \quad b_i = (i-1)\frac{\pi}{4}.$$

The probabilistic description of the source is

$$p(A_k) = 1/4, \qquad \text{for all } A_k$$

$$p(\theta_k/A_k = \alpha_1) = \begin{cases} 1/4, & \theta_k = \beta_2, \beta_4, \beta_6, \beta_8 \\ 0, & \text{otherwise} \end{cases}$$

$$p(\theta_k/A_k = \alpha_3) = p(\theta_k/A_k = \alpha_1)$$

$$p(\theta_k/A_k = \alpha_2) = \begin{cases} 1/4, & \theta_k = \beta_1, \beta_3, \beta_5, \beta_7 \\ 0, & \text{otherwise} \end{cases}$$

$$p(\theta_k/A_k = \alpha_4) = p(\theta_k/A_k = \alpha_2). \tag{41}$$

In place of the maximization problem posed in (23), we write

$$\max_{\{\psi_k\}_1^K, \{\Delta\theta_k\}_1^K, \{A_k\}_1^K} f\left(\{x_k\}_1^K, \{\psi_k\}_1^K, \{\Delta\theta_k\}_1^K, \{A_k\}_1^K\right) \tag{42}$$

with ψ_k and $\Delta\theta_k$ defined as in (22). The density $f^K(\cdot, \cdot, \cdot, \cdot)$ appearing in (42) may be written

$$f^K = \prod_{k=1}^{K} N_{x_k}\left(A_k e^{j\psi_k}, \sigma_n^2\right) \cdot f\left(\psi_k, \Delta\theta_k, A_k / \{\psi_j\}_1^{k-1}, \{\Delta\theta_j\}_1^{k-1}, \{A_j\}_1^{k-1}\right). \tag{43}$$

The conditional density on the right-hand side of (43) is simply

$$f(\psi_k, \Delta\theta_k, A_k/\cdot, \cdot, \cdot) = g_1(\psi_k - \psi_{k-1} - \Delta\theta_k)\, p(\Delta\theta_k/A_k, A_{k-1})\, p(A_k) \tag{44}$$

where $p(\Delta\theta_k/A_k, A_{k-1})$ is the conditional probability mass function for $\Delta\theta_k$, given A_k *and* A_{k-1}. Putting (43) and (44) together, we have as the joint density function to be maximized

$$f^K = \prod_{k=1}^{K} N_{x_k}\left(A_k e^{j\psi_k}, \sigma_n^2\right) g_1(\psi_k - \psi_{k-1} - \Delta\theta_k) \cdot p(\Delta\theta_k/A_k, A_{k-1})\, p(A_k). \tag{45}$$

It is important to note in this expression that the $N_{x_k}(\cdot, \cdot)$ term is dependent only on the measurement model; $g_1(\cdot)$ is dependent only on the random phase model, and $p(\Delta\theta_k/\cdot, \cdot)p(A_k)$ is dependent only upon the symboling constellation (or encoding scheme). Thus (45) is a useful canonical decomposition that is generally applicable to communications problems involving additive independent noise and independent increments phase processes.

For the (4, 4) diagram of Fig. 2(d) we may compute $p(\Delta\theta_k/A_k A_{k-1})$ as follows:

$$p(\Delta\theta_k/A_k = \alpha_i, A_{k-1} = \alpha_j) = \begin{cases} 1/4, \ \Delta\theta_k = \beta_1, \beta_3, \beta_5, \beta_7, \\ \quad i, j \text{ even-even or odd-odd} \\ 1/4, \ \Delta\theta_k = \beta_2, \beta_4, \beta_6, \beta_8, \\ \quad i, j \text{ even-odd or odd-even} \end{cases} \tag{46}$$

It is a straightforward matter to substitute these results into (45) and derive a path metric as in (37).

VI. Linear Performance Results and the Selection of a Fixed Lag

There is one more simplification to be made: namely, the selection of a depth constant k_0 such that phase-symbol pairs may be decoded at a fixed-lag k_0, thereby obviating the need to store long survivor sequences. Call $\{\hat{\psi}_{k/K}\}_1^K$ the MAP phase sequence based on measurements $\{x_k\}_1^K$. The subscript k/K indicates that $\hat{\psi}_{k/K}$ depends on all measurements up to time K. In general the MAP sequence $\{\hat{\psi}_{k/K+1}\}_1^{K+1}$ based on measurements to time $(K+1)$ may differ from $\{\hat{\psi}_{k/K}\}_1^K$ at all values of $1 \le k \le K$. However, one expects that for large K and for $k \le K - k_0$, the sequences $\{\hat{\psi}_{l/K}\}_1^k$ and $\{\hat{\psi}_{l/K+1}\}_1^k$ will not be very different for a well-chosen depth k_0. In other words, long survivor sequences tend to have one common trunk up to $K - k_0$, at which point they may diverge as illustrated in Fig. 6. Thus we may use $\hat{\psi}_{K-k_0/K}$ as a final estimate of $\hat{\psi}_{K-k_0}$ since $\hat{\psi}_{K-k_0/K+l} \doteq \hat{\psi}_{K-k_0/K}$ for all positive l. Thus as a practical matter, one may choose a depth constant k_0 such that the sequence of fixed-lag estimates $\hat{\psi}_{k-k_0/k}$, $k = k_0 + 1, k_0 + 2, \cdots$, gives an approximate MAP sequence. Here $\hat{\psi}_{k-k_0/k}$ is simply the phase value, k_0 samples back, in the MAP sequence based on measurements up to time k. In this way, phase values are estimated with delay k_0 and only survivor sequences of length k_0 must be stored.

How should k_0 be chosen? This is a difficult question to answer precisely, because no analytical results exist for the performance of nonlinear phase trackers of the Viterbi-type. We can, however, study the filtering behavior of a related linear problem and find how performance varies with

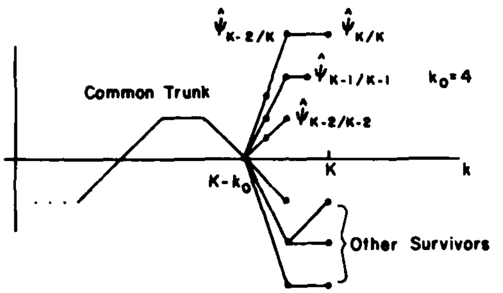


Fig. 6. Illustration of survivor evolution with common trunk.

fixed-lag k_0. To this end, we consider the problem of tracking phase when there is no data symboling. Assume $\{\psi_k\}$ is a normal random walk of the form (6) with $w_k : N_{w_k}(0, \sigma_w^2)$. Let $x_k = e^{j\psi_k} + n_k$, $\{n_k\}$ be a sequence of complex random variables whose real and imaginary parts are i.i.d. $N_{n_k}(0, \sigma_n^2)$ random variables. A PLL with gain K_1 for estimating $\{\psi_k\}$ is the following:

$$\hat{\psi}_k = \hat{\psi}_{k-1} + K_1 \,|x_k|\sin\left(\arg x_k - \hat{\psi}_{k-1}\right). \quad (47)$$

Note that this is similar to (13) when there is no data.

For $\sigma_n^2 \ll 1$ we approximate (47) with

$$\hat{\psi}_k = \hat{\psi}_{k-1} + K_1\left(\arg x_k - \hat{\psi}_{k-1}\right). \quad (48)$$

When K_1 is selected to be

$$K_1 = \left(\sigma_w^2/\sigma_n^2\right)\left[-0.5 + 0.5\left(1 + 4\sigma_n^2/\sigma_w^2\right)^{1/2}\right], \quad (49)$$

then (48) is the Kalman filter for the "linear observation model"

$$\arg x_k = \psi_k + n_k \to x_k = \exp\left[j(\psi_k + n_k)\right]. \quad (50)$$

The steady-state filtering error P_0 for this linear problem is related to K_1 as follows:

$$K_1 = \frac{\sigma_w^2}{\sigma_n^2}\cdot\frac{P_0}{\sigma_w^2}. \quad (51)$$

A general result due to Hedelin [12] for fixed-lag smoothing may be adapted to random walk smoothing from observations of the form (50). The steady-state fixed-lag smoothing variance P_{k_0} at delay k_0 is

$$\begin{aligned} P_{k_0}/\sigma_w^2 &= P_0/\sigma_w^2 - \sum_{l=1}^{k_0} G^{2l} \\ &= P_0/\sigma_w^2 - G^2(1 - G^{2k_0})/(1 - G^2) \\ G &= 1 - K_1. \end{aligned} \quad (52)$$

The infinite-lag smoothing variance is

$$P_\infty/\sigma_w^2 = P_0/\sigma_w^2 - G^2/(1 - G^2). \quad (53)$$

In Fig. 7 several error expressions and asymptotic forms are plotted versus σ_w^2/σ_n^2, which is a kind of SNR. For large σ_w^2/σ_n^2, the error variances P_0/σ_w^2, P_{10}/σ_w^2, and P_∞/σ_w^2 go as $(\sigma_w^2/\sigma_n^2)^{-1}$. For small σ_w^2/σ_n^2, they go as $(\sigma_w^2/\sigma_n^2)^{-1/2}$ although infinite-lag smoothing offers 6 dB improvement in σ_w^2/σ_n^2 over zero-lag smoothing for a fixed smoothing variance. Over the range of values $0.01 \le \sigma_w^2/\sigma_n^2 \le 10$, a delay of $k_0 = 10$ offers all but 1–2 dB of the theoretically achievable gain from infinite delay. In communication problems for which random phase is a significant effect, the ratio σ_w^2/σ_n^2 is typically in this range. Only at very small values of σ_w^2/σ_n^2 can very large delays k_0 provide large performance gains, but in this case there is no real phase fluctuation problem for the purpose of data decoding, and the gain is not worth the large delay. Shown also in Fig. 7 is the Kalman gain K_1 versus σ_w^2/σ_n^2.

The problem considered in Section IV is admittedly different from the linear problem considered here. However, the numerical results given in Fig. 8 for the Viterbi phase tracker illustrate that the performance gain to be achieved with a fixed-lag of $k_0 = 10$ is much as predicted by the linear theory. For the results of Fig. 8, the phase space was discretized to $m = 48$ values, data transmission was 8-ary PSK, and the decoding algorithm was the VA. The circles, dots, and squares represent experimental phase estimation error variances, and the heavy solid lines represent theoretical results. Over the range of values $0.1 \le \sigma_w^2/\sigma_n^2 \le 2$, the phase estimator variance for the Viterbi phase tracker operating with delay $k_0 = 10$ is essentially equivalent to the filtering variance of a Kalman filter that has access to linear observations and provides estimates without delay. Performance is not measurably degraded by the presence of data which are concurrently decoded.

VII. Simulation Results: Gaussian Increments

For all simulation results discussed in this section the phase space $[-\pi, \pi)$ has been discretized to 48 equally spaced phase values and a Viterbi algorithm has been programmed to solve the MAP sequence estimation problem. The principle of optimality established in Section V has been used to derive the appropriate path metric and thereby reduce computational complexity. The choice of a fixed-lag decoding (or depth) constant is $k_0 = 10$. Source symbols have been generated independently. The random phase sequence has been governed by the independent increments model of (6) with $w_k : N(0, \sigma_w^2)$ and initial phase uniformly distributed on $[-\pi, \pi)$. Initial phase acquisition has been achieved by transmitting a preamble according to one of the following schemes.

a) During a pretransmission period of length N, the sequence of transmitted data is known to the receiver. Thus in the DBVA and VA systems, based upon MAP estimation, the Viterbi algorithm works as a pure phase estimator during this period. At the end of the preamble, the Viterbi algorithm is turned into a joint phase-data MAP estimator. In the DDPLL and JE systems, based upon decision-directed algorithms, the algorithm is directed by the true data during the preamble period.

b) During the preamble period, identical (but unknown) data are transmitted. This keeps the phase from making phase jumps associated with symbol changes and makes the joint phase-data estimator able adequately to acquire the initial phase.

In our simulations the VA has achieved the same data-error probability for both methods; i.e., its performance

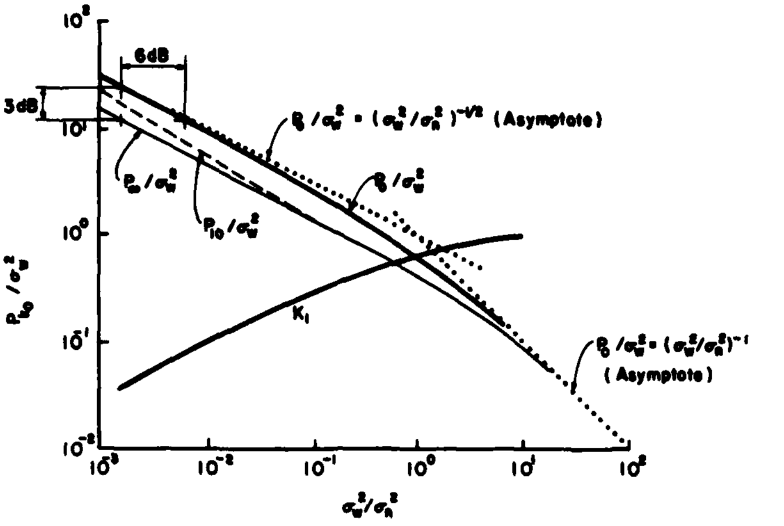


Fig. 7. Linear performance results for evaluating effects of fixed-lag k_0.

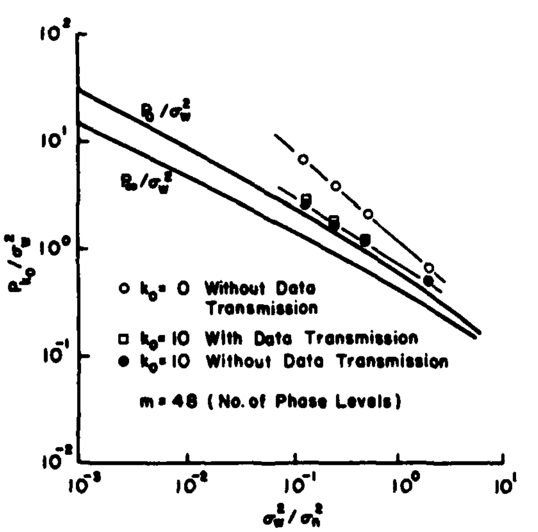


Fig. 8. Selected phase tracking variances with and without data transmission.

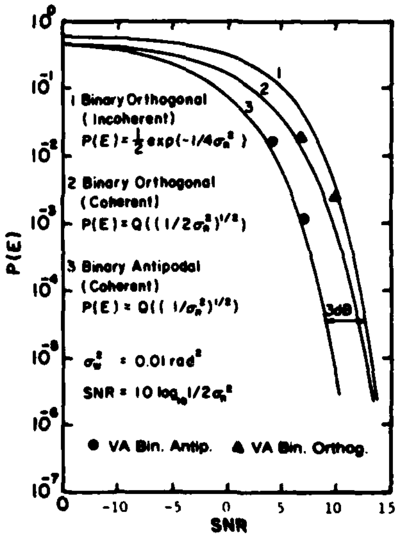


Fig. 9. Symbol error probabilities for binary symboling.

has not depended upon which learning procedure was used. On the other hand, Ungerboeck's DBVA have proved to be sensitive to the learning procedure. For example, at SNR = 20 dB with phase variance $\sigma_w^2 = 4\,\sigma_n^2$ for a learning period of $N = 60$ data, the number of errors during a transmission period of 490 data values has jumped from seven for procedure a)—known data—to 59 for procedure b)—constant but unknown data. Moreover, the DBVA typically requires a longer learning period than does the VA (roughly two times longer). A value of $N = 50$ is sufficient for the VA, while the DBVA needs $N = 100$ learning iterations in our simulations. The decision-directed systems (DDPLL and JE) work as the VA in these respects. That is, a preamble period of 50 data values is sufficient. These data may be unknown to the receiver, provided they are kept constant (procedure b)). No degradation with respect to procedure a) results.

Binary Symboling

Shown in Fig. 9 are binary symboling results for the VA when $\sigma_w^2 = 0.01\,\text{rad}^2 (\sigma_w = 5.7°)$ and SNR ranges from 4 to 10 dB. (Recall SNR $= 10 \log_{10} 1/2\sigma_n^2$.) The results indicate that performance with the VA is essentially equivalent to that of a fully coherent receiver, even for a relatively large value of σ_w. For comparison, the curves for coherent binary orthogonal and coherent binary antipodal systems are also shown. The simulation results for binary orthogonal symboling are interesting because they serve to validate the simulation. Indeed, as expected, the performance of the VA is seen in Fig. 9 to lie between that of an incoherent receiver and that of a fully coherent receiver. Of course, the margin between coherent and incoherent performance is small at SNR's of practical interest. The simulation results for binary antipodal symboling are interesting on their own

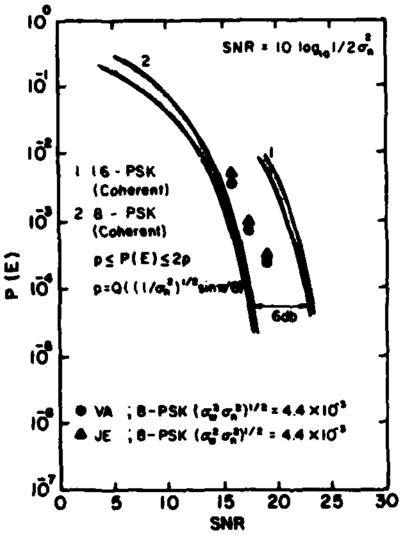


Fig. 10. Symbol error probabilities for eight-PSK.

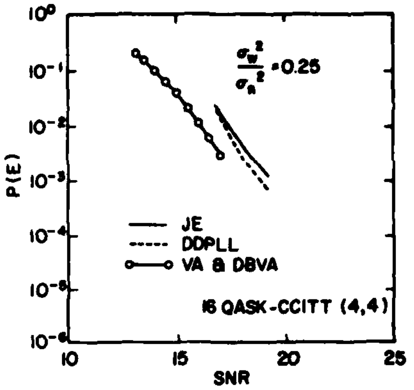


Fig. 11. Symbol error probabilities for 16-QASK-CCITT symboling (low phase noise).

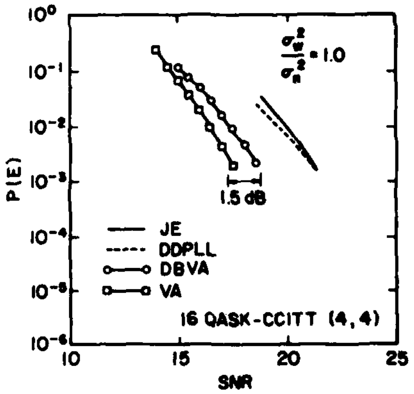


Fig. 12. Symbol error probabilities for 16-QASK-CCITT symboling (average phase noise).

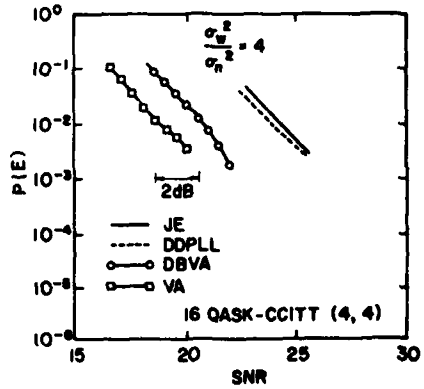


Fig. 13. Symbol error probabilities for 16-QASK-CCITT symboling (high phase noise).

because incoherent reception is not possible with antipodal symboling.

Eight-PSK

Shown in Fig. 10 are simulation results for eight-PSK when SNR ranges from 16–19 dB and $(\sigma_w^2\sigma_n^2)^{1/2}$ remains fixed at 4.4×10^{-3} rad^2. This choice of parameters corresponds to a loop SNR of 23 dB where the PLL is well into its linear region of operation and little can be gained from improvements to the phase tracking. The values of σ_w^2 under investigation range from 1.6° to 2.2° and the ratio σ_w^2/σ_n^2 is very small, ranging from 0.03 to 0.12. The solid circles of Fig. 10 correspond to the VA, and the solid triangles correspond to the markedly simpler JE. Also shown in Fig. 10 are performance bounds for fully coherent eight-PSK and 16-PSK symboling. In this case neither the VA nor the DBVA provides significant improvement over the JE or DDPLL. The latter two receivers are simpler than the DBVA which, in turn, is simpler than the VA. Therefore, for such cases of weak phase noise, neither the VA nor the DBVA would be favored over the JE or the DDPLL.

16-QASK

Shown in Fig. 11–13 are simulation results for 16-QASK symbols encoded according to the (4, 4) CCITT rule. The decoding procedure are JE, DDPLL, DBVA, and VA, for three distinct values of the ratio σ_w^2/σ_n^2. Fig. 11 is concerned with a weak phase noise ($\sigma_w^2/\sigma_n^2 = 0.25$). Fig. 12 is concerned with an average phase noise ($\sigma_w^2/\sigma_n^2 = 1$), and Fig. 13 is concerned with a large phase noise ($\sigma_w^2/\sigma_n^2 = 4$). We recall [1] that the DBVA performs some kind of phase estimation along a path that satisfies

$$\hat{\psi}_n = \hat{\psi}_{n-1} \pm \sigma_w, \qquad (54)$$

using a Viterbi algorithm. The DBVA that we have simulated is somewhat different from Ungerboeck's DBVA, in which the number of possible phase states at each iteration is limited to six or eight. In our simulation the number of phase states is not limited, thus avoiding one possible cause of errors and improving the error rate, but also increasing the computational complexity with respect to [1].

Behavior of DDPLL and JE on CCITT (4, 4) Constellation

The decision-directed algorithms (DDPLL and JE) have essentially the same performance, as shown in Figs. 11–13. The DDPLL is superior to the JE by only 0.5 dB. The slight inferiority of the JE is largely compensated by the fact that the complex gain of the JE can also correct rapid gain fluctuations in the channel. We emphasize that the curves of the DDPLL and JE are biased and cannot be trusted just as they are because of the occurrences of very large bursts of errors at relatively high error probabilities. When such bursts have occurred in the simulation runs, they have been withdrawn from the error rate computation. For instance, with $\sigma_w^2 = 0.25\sigma_n^2$ and SNR = 17 dB, at an error probability on the order of 10^{-2}, between one fourth and one third of the simulation runs (with length 500 data values) have exhibited bursts of about a hundred errors. In the simulations, the bursts began to occur at SNR = 18 dB, 21.5 dB, and 26 dB for $\sigma_w^2/\sigma_n^2 = 0.25$, 1, and 4, respectively. This corresponds to a value of σ_w such that $4\sigma_w$ ranges between 11.5° and 20°. The phenomenon of error bursts can be explained as follows: because the phase increment is Gaussian it will occasionally reach the value $4\sigma_w$. If, at the same time, the noise is relatively large, the angle between the observed data and the transmitted symbol will exceed the value 22.5° that corresponds to the angular threshold for an error in the 16-point CCITT diagram (see Figs. 2(d) and 4(a)). No type of decision-directed phase estimator can correct such an error. Therefore, the phase estimate will become incorrect (by a shift of ±45°), causing a group of errors. In turn, due to the decision-directed nature of the phase estimator, error multiplication occurs, resulting in an error burst. The importance of the burst phenomenon in the decision-directed algorithms can be appreciated from Table I. The table gives observed burst frequency in runs of 500 samples, parameterized by the corresponding observed probability of an isolated error. The results are given for a DDPLL, but they are essentially the same for the JE. Moreover, the results are relatively independent of the ratio σ_w^2/σ_n^2 in the range 0.25–4. A decision-directed algorithm is not a reliable phase estimator when the error probability reaches the level of 10^{-2}, corresponding to severe transmission channels. With respect to burst phenomena, the DDPLL and JE behave similarly.

Behavior of DBVA and VA on CCITT (4, 4) Constellation

The performance of the VA is superior to that of the DBVA. The gain achieved by the VA over the simpler DBVA is monotone increasing in the ratio of phase fluctuation variance σ_w^2 to additive noise variance σ_n^2. While there is no gain when $\sigma_w^2/\sigma_n^2 = 0.25$, the gain is 1 dB for $\sigma_w^2/\sigma_n^2 = 1$ and 2 dB for $\sigma_w^2/\sigma_n^2 = 4$. Both systems perform better than the DDPLL or JE, the improvement again being a monotone increasing function of σ_w^2/σ_n^2.

A very important point is that the use of either of the two MAP phase estimators precludes the occurrence of error bursts. The errors seem to be grouped in twos or threes, and no error multiplication occurs since the phase estimator is not decision-directed. Thus such MAP sequence estimators can be used even at high error probabilities on the order of 10^{-2} or 10^{-1}.

TABLE I
OBSERVED BURSTS

P_E (Isolated Error)	$< 10^{-4}$	2.0×10^{-3}	1.5×10^{-2}
Percent bursts	0	7	20

Comparison Between MAP and Decision-Directed Phase Estimators

The improvement that can be gained by using any type of MAP estimator for phase rather than a simple decision-directed algorithm is again an increasing function of σ_w^2/σ_n^2. Fig. 11 shows that only 1 dB is gained by the DBVA and the VA over the DDPLL if $\sigma_w^2 = 0.25\ \sigma_n^2$. This gain is realized at a high computational price. For the phase fluctuations and additive noise of the same importance ($\sigma_w^2/\sigma_n^2 = 1$), the VA outperforms the DDPLL by 3 dB (see Fig. 12), but the gain is reduced to 2 dB for the simpler DBVA. For large phase fluctuations, the gain is important. For instance, Fig. 13 shows that the VA outperforms the DDPLL by 5 dB when $\sigma_w^2/\sigma_n^2 = 4$. In addition, the VA brings the insurance that no burst of errors can occur, even for very poor SNR and large phase fluctuations. In fact, the true power gain of the VA over a decision-directed algorithm is even higher than just claimed if one takes account of the additional power required in the decision-directed schemes to ensure against burst as well as random errors.

Sensitivity to Imperfect Knowledge of σ_w^2/σ_n^2

It is easily seen in (18) or (37) that the only parameter required in order to proceed with the VA algorithm is the ratio of phase variance to additive noise power. The same holds for the DDPLL whose optimal gain K_1 depends on this ratio (see (49)), and for the JE whose step-size μ (see (14)) is to be kept close to K_1, but smaller, provided the data diagram has unit power. As for the DBVA, it requires only the knowledge of σ_w^2 in order to determine the number m of discretized phase levels. Thus an important feature of each system is its sensitivity to an imperfect knowledge of σ_w^2/σ_n^2 (or σ_w^2) because, first, σ_w^2 can vary with time and, second, the actual phase can fluctuate according to a statistical model that is different from the one expected. The less sensitive the system is to the knowledge of σ_w^2/σ_n^2 (or σ_w^2), the more robust it is.

a) Sensitivity of the Decision-Directed Systems: Let us denote σ_w^2/σ_n^2 by α. The function $K_1(\alpha)$ that gives the optimum loop-gain of the DDPLL is sketched in Fig. 14. It is quite flat except for α very close to zero (e.g., $\alpha < 0.2$).

Now the case $\alpha \ll 1$ is of no real interest for the purpose of this paper. Indeed, it has been seen previously that, in

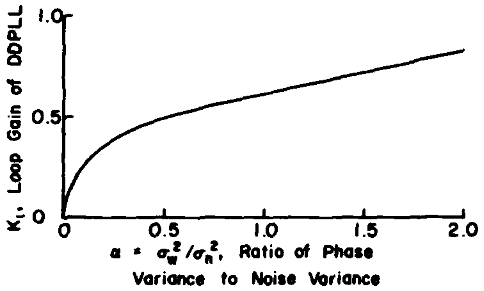


Fig. 14. Optimum gain-loop of DDPLL.

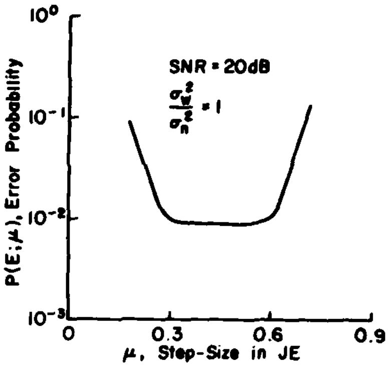


Fig. 15. Sensitivity of JE to choice of step size μ.

this case, no MAP phase estimator is worth being worked out. Moreover, any reasonable phase estimator will perform satisfactorily. When α is not negligible, $K_1(\alpha)$ is slowly varying. For example, $K_1(1)/K_1(0.25) = 1.59$, and $K_1(4)/K_1(1) = 1.34$. Thus the value $K_1(1) = 0.62$ for the DDPLL gain is correct for a large range of values of α. This fact is largely confirmed by the simulations. Hence, due to the risk of error multiplication that increases very rapidly with K_1, it should rather be set to the lower bound $K_1(\alpha_{\min})$ corresponding to the smallest α that can be expected, rather than to an average value $K_1(\alpha_{\text{ave}})$, which will sometimes be too large and bring error bursts. Thanks to this precaution, the DDPLL is insensitive to α. It is a robust system.

The robustness of the JE is also excellent. This fact was checked on numerous computer simulations: as a function of the step size μ, the error probability $P(E; \mu)$ exhibits a minimum which is very flat, as sketched in Fig. 15. The range where the minimum is reached does not depend critically upon α. A value such as $\mu = 0.4$ corresponds to the minimum of error probability for α in the range [0.25–1] and for a unit energy data diagram.

b) Sensitivity of the MAP Phase Estimators: The VA sensitivity to imperfect knowledge of α has been tested in our computer simulations. It appears that the VA performance is not appreciably degraded by an error of ± 6 dB for α. Hence the VA robustness is at least as good as that of the decision-directed algorithms.

On the other hand, the DBVA robustness has turned out to be poor. For instance, with SNR = 21 dB and $\alpha = 4$, the DBVA is supposed to work with $m = 2\pi/\sigma_w = 50$ phase levels. If only 45 levels are used, corresponding to a 0.9 dB error for α, then the error probability is increased by a factor of two. In fact, as a function of m, $P(E; m)$ exhibits a minimum, but it is a sharp minimum. This poor robustness can be understood by noting that in the DBVA, the path metric is not a function of $\alpha = \sigma_w^2/\sigma_n^2$, but only of σ_w^2. This may be one of the main drawbacks of the DBVA.

VIII. Simulation Results: Bounded-Increments Phase Jitter

For all simulation results of this section the phase space $[-\pi, \pi)$ has been discretized to 32 equally spaced phase values, and a VA has been programmed to solve (17). The assumed increment density $h(w)$ is the uniform density

$$h(w) = \begin{cases} \dfrac{1}{2a}, & -a \le w < a;\ 2a = 2\pi/16 \\ 0, & \text{otherwise.} \end{cases} \tag{55}$$

The corresponding discrete transition density for use in the path metric is

$$f(\phi_k/\phi_{k-1}) = \begin{cases} 1/3, & \phi_k - \phi_{k-1} = -\pi/16, 0, \pi/16 \\ 0, & \text{otherwise.} \end{cases} \tag{56}$$

The resulting VA is related to the class of so-called M algorithms [16]–[18] in which all survivors are saved, but only M (in this case 3) candidate originator states are allowed. This significantly reduces calculations and results in an algorithm similar in spirit to the DBVA of [1]. Still, however, phase is tracked only on $[-\pi, \pi)$ rather than on $(-\infty, \infty)$.

Source symbols have been generated independently from a four-PSK alphabet and used to differentially encode phase according to a Gray code. The random phase sequence has been generated in ways to be discussed below.

Markov Phase with Non-Gaussian Increments

Here the phase is generated according to (6) with $h(w)$ given by (55). Thus the algorithm is matched to the actual phase sequence. Shown in Fig. 16 are performance results for the VA and for the JE. The VA outperforms the JE by 1.5 dB over the range 10 dB ≤ SNR < 15 dB. The probability of error is "probability of bit error."

Sinusoidal Phase Jitter

Here the phase jitter is sinusoidal (see (4)) with uniformly distributed initial phase and frequency ν. The frequency is chosen such that $\nu\Delta = 1/24$, corresponding to a transmission rate of 4800 bits/s with baud rate $1/\Delta =$ 2400 Hz and jitter frequency $\nu = 100$ Hz. The runs are 2000–10,000 steps long, corresponding to 4000–20,000 transmitted bits. The peak-to-peak phase deviation is 20° or 60°. For these experiments the VA outperforms the JE by 1.5–1.7 dB. This gain is, of course, achieved at a high price in complexity.

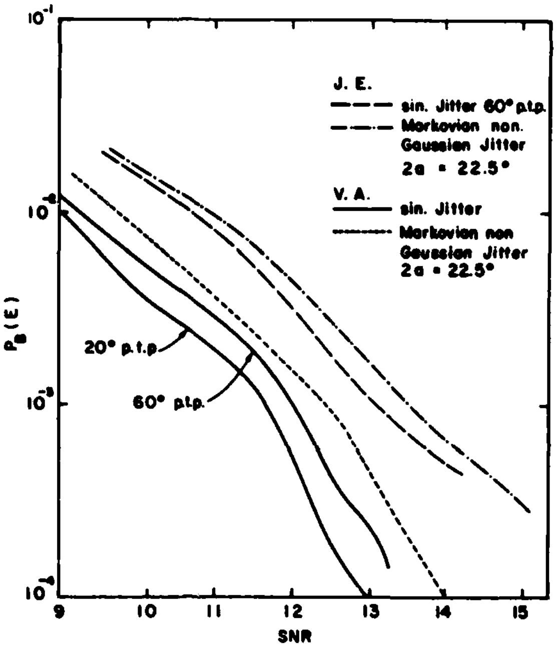


Fig. 16 Symbol error probabilities for 4-PSK and non-Gaussian phase increments.

Comparison of the JE and VA

In the simulations reported above, the ratio $\alpha = \sigma_w^2/\sigma_n^2$ ranges from 0.02 to 0.81, that is from small to average values. No burst of errors has ever been observed for the JE. This is due to the fact that the phase increment is always bounded as appears in (4) and also (55). The bound is much smaller than the angular distance between adjacent data. Thus there is no risk of a ±90° slip (corresponding to the four-PSK diagram) in the JE phase estimation. Hence the errors will be scattered rather than grouped, and no error multiplication phenomenon can happen.

Owing to this consideration, to the fact that the VA outperforms the JE by only 1.5 dB, and to the complexity of the VA, a practical system will implement the JE (or DDPLL) rather than the VA (or DBVA), in the case of bounded increment phase jitter.

IX. Conclusion

We have derived a principle of optimality for phase-amplitude encoded symboling that allows one to simultaneously track random phase and decode data symbols using the VA derived in [8] and [9]. The VA is designed for a random walk phase process, a very severe type of phase process. In such a process there exists the possibility of large phase jumps. The VA gives excellent performance because it benefits from the use of a lag to observe future data samples which make large phase jumps look unlikely.

In order to reach conclusions about the type of phase estimation that should be used for given types of phase fluctuations, performance comparison of the VA with two simple decision-directed (zero-lag) phase estimators, namely, the JE of [3] and the DDPLL of [5], and with the DBVA of [1], have been thoroughly investigated by computer simulations, with various data diagrams. They indicate that the choice among the four systems is to be made according to four parameters:

1) the error probability $P(E)$ at which the system is to be used:
2) the relative importance $\alpha = \sigma_w^2/\sigma_n^2$ of phase fluctuations with respect to additive noise;
3) the complexity c that is technologically feasible and acceptable;
4) the maximum phase increment $\Delta\Phi_{\max}$ that is to be expected, as compared to the angular distance between points of the data diagram.

TABLE II
Choice of Phase Tracker

	$P(E)$	Small	Large
α			
Small		*case 1* JE or DDPLL	*case 2* see Table III
Large		*case 3* see Table III	*case 4* VA or DBVA

TABLE III
Choice of Phase Tracker: Continued

	c	Small	Large
$\Delta\Phi_{\max}$			
Small		JE or DDPLL	JE or DDPLL
Large		DBVA	VA or DBVA

Suggestions for this choice are sketched in Tables II and III where Table III is concerned with cases 2 and 3 of Table II.

The choice between the two decision-directed phase estimators, JE or DDPLL, is irrelevent for the matters discussed in this paper. It appears in Tables II and III that the VA and DBVA are preferred when α, $P(E)$, and $\Delta\Phi_{\max}$ are large. The comparison between these two MAP phase estimators shows that the VA is more robust, has a smaller learning period, and outperforms the DBVA by 2 dB or more when α is at least equal to four.

Only Viterbi, or Viterbi-like, algorithms can survive and correct error bursts by effectively using the weight of future evidence to render such bursts too unlikely to occur. Thus it seems likely that the VA (really dynamic programming) will grow in importance in such applications as spread spectrum communication where the phase of a wide-band carrier can be tracked for symbol decoding.

Acknowledgment

The authors wish to thank S. Kerbrat and C. Pariente for their assistance with software development and simulations.

Appendix
Monotonicity of Folded Normal and Cauchy Densities

There are many choices for the phase increment density $h(x)$ that are physically interesting and mathematically tractable. Two of particular interest are the normal density and the Cauchy, the latter being useful in the modelling of "heavy-tailed" behavior. When folded around the unit circle according to (8) these densities yield transition densities which achieve their maximum at $\phi_k - \phi_{k-1} = 0$ and decrease monotonically on the interval $0 \leq \phi_k - \phi_{k-1} \leq \pi$.

Consider first the Cauchy case

$$g_1(x) = \sum_{k=-\infty}^{\infty} \frac{a/\pi}{a^2 + (x + k2\pi)^2}. \qquad (57)$$

According to Poisson's summation formula [13], this may be written

$$\begin{aligned} g_1(x) &= \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} e^{-a|k|} e^{jkx} \\ &= \frac{1}{2\pi}(1 - e^{-2a})(1 - 2e^{-a}\cos x + e^{-2a})^{-1}. \qquad (58) \end{aligned}$$

This function achieves its maximum value at zero and decreases monotonically.

In the normal case

$$g_1(x) = \sum_{k=-\infty}^{\infty} (2\pi\sigma^2)^{-1/2} \exp\left\{-(x + 2k\pi)^2/2\sigma^2\right\}. \qquad (59)$$

Again, by Poisson's summation formula,

$$g_1(x) = \sum_{k=-\infty}^{\infty} (2\pi)^{-1} \exp\left\{jkx - k^2\sigma^2/2\right\}. \qquad (60)$$

This infinite sum goes by the name $J_3(x, q = e^{-\sigma^2/2})$ in the theory of Jacobian elliptic functions and theta functions [15]. The theta function $J_3(x, q)$ is known to be monotonically decreasing on the interval $0 \leq x \leq \pi$.

References

[1] G. Ungerboeck, "New application for the Viterbi algorithm: Carrier phase tracking in synchronous data transmission systems," in *Proc. Nat. Telecomm. Conf.*, 1974, pp. 734–738.

[2] H. Kobayaski, "Simultaneous adaptive estimation and decision algorithm for carrier modulated data transmission systems," *IEEE Trans. Commun. Technol*, vol. COM-19, pp. 268–280, June 1971.

[3] M. Levy and O. Macchi, "Auto-adaptive phase jitter and interference intersymbol suppression for data transmission receivers," presented at the *Nat. Telecomm. Conf.*, Dallas TX; Nov. 1976.

[4] ——, "Egaliseur de Gigue," *6ème Colloque GRETSI sur le Traitement du Signal et ses Applications*, Nice, France, April 1977.

[5] D. D. Falconer, "Analysis of a gradient algorithm for simultaneous passband equalization and carrier phase recovery," *Bell Syst. Tech. J.*, vol. 55, pp. 409–428, 1976.

[6] F. R. Magee, "Simultaneous phase tracking and detection in data transmission over noisy dispersive channels," *IEEE Trans. Commun.* vol. COM-25, pp. 712–715, July 1977.

[7] A. S. Willsky, "Fourier series and estimation on the circle with applications to synchronous communications—Part I: Analysis," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 577–583, Sept. 1974.

[8] L. L. Scharf, D. D. Cox, and C. J. Masreliez, "Modulo-2π phase sequence estimation," *IEEE Trans. Inform. Theory*, vol. IT-26, pp. 615–620, Sept. 1980.

[9] L. L. Scharf, "A Viterbi algorithm for modulo-2π phase tracking in coherent data communication systems," ONR Tech. Rep. 25, Feb. 1978.

[10] G. D. Forney, "Maximum likelihood sequence estimation of digital sequences in the presence of intersymbol interference," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 363–378, May 1972.

[11] G. J. Foschini, R. D. Gitlin, and S. B. Weinstein, "On the selection of a two-dimensional signal constellation in the presence of phase jitter and Gaussian noise," *Bell Syst. Tech. J.*, vol 52, pp. 927–965, July–Aug. 1973.

[12] P. Hedelin, "Optimal smoothing in discrete continuous linear and nonlinear systems," *Inform. Sci.*, vol. 13, pp. 137–158, 1977.

[13] S. Bochner, *Harmonic Analysis and The Theory of Probability*. Univ. of California Press, 1960, Berkeley, CA: pp. 28–36.

[14] *Data Communications Using the Switched Telecommunications Network*, Bell Syst. Tech. Ref., pp. 15–16, Aug. 1970.

[15] M. Abramowitz and I. Stegun, *Handbook of Mathematical Functions*. National Bureau of Standards: U.S. Government Printing Office, pp. 567–578, 1970.

[16] F. Jelinek and J. B. Anderson, "Instrumentable tree encoding of information sources," *IEEE Trans. Inform. Theory*, vol. IT-17, pp. 118–119, Jan. 1971.

[17] S. G. Wilson and D. W. Lytle, "Trellis encoding of continuous-amplitude memoryless sources," *IEEE Trans. Inform. Theory*, vol. IT-23, pp. 404–409, May 1977.

[18] G. Kawas-Kaleh, "Réception de données par décodage récurrent en présence de bruit et d'interférences entre symboles," Thèse ingenieur docteur, Paris, France, Mar. 1977.

Aspects of Dynamic Programming in Signal and Image Processing

LOUIS L. SCHARF, SENIOR MEMBER, IEEE, AND HOWARD ELLIOTT, MEMBER, IEEE

Abstract—The techniques peculiar to dynamic programming have found a variety of successful applications in the theory and practice of modern control. Successes in the theory and practice of signal and image processing are less numerous and prominent, but they do exist. In this paper, we sound a call for renewed attention to the potential of dynamic programming for solving knotty nonlinear filtering problems in signal and image processing, and outline successes we have recently enjoyed in nonlinear frequency tracking and random boundary estimation in noisy black and white images. Two classical results, the fast Fourier transform and Levinson's recursion for determining autoregressive parameters, are treated in the context of dynamic programming simply to reinforce the point that many of the algorithms we take for granted, and which were derived without recourse to dynamic programming, can be nicely interpreted as dynamic programming algorithms.

I. Introduction

IN THIS PAPER it is our aim to show that dynamic programming, a fundamental technique in control theory since Bellman's introduction and advocacy of it in the mid-1950's, can be of considerably more value in signal and image processing than has generally been recognized. This is not to say others have failed to recognize the potential of dynamic programming for solving interesting signal processing problems. We mention in particular Cox's early work [1], [2] on Kalman filtering and dynamic programming for the estimation of state variables and the identification of system parameters; Viterbi's dynamic programming algorithm for decoding convolutional code sequences [3]; Cahn's dynamic programming algorithm for FM demodulation [4]; and Forney's discussion of inference problems on finite-state Markov sequences that can be solved with the techniques of dynamic programming [5].

In the sections to follow we rederive classical algorithms in discrete Fourier analysis and linear prediction using the principle of dynamic programming. We then present two new dynamic programming algorithms. One is for nonlinear frequency tracking and the other is for edge detection in noisy black and white images.

The organization is as follows. In Section II, we present an elementary dynamic programming formalism. In Section III we use dynamic programming arguments to rederive the Goertzel and decimation-in-frequency fast Fourier transform (FFT) algorithms for efficiently computing the discrete Fourier transform (DFT). In Section IV, we discuss the connections between control, detection, estimation, and prediction of autoregressive sequences observed in additive noise. We highlight the central role played by the so-called normal equations and rederive the Levinson algorithm for recursively solving them in the order of p^2 operations. The derivation is a dynamic programming one.

The new results follow in Sections V and VI. In Section V, a dynamic programming algorithm for tracking the frequency of a frequency modulated sequence in additive noise is derived. Several simulations illustrate the performance of the algorithm. This provides a solution to a classical nonlinear filtering problem. The results of Section VI show how dynamic programming may be used to derive a new algorithm for estimating local segments of object boundaries in noisy black-and-white images. Some examples are given to illustrate the use of the algorithm in estimating complete object boundaries as well.

II. A Dynamic Programming Formalism

Traditionally, dynamic programming has been used to find "optimu·n" solutions to multistage decision problems [6], [7]. An "optimum" solution has generally been one that maximizes or minimizes a performance or cost functional. When the multi-stage decision problem is cast in a probabilistic framework and the criterion of optimality is maximum *a posteriori* (MAP) probability, then the cost functional is typically a multivariable likelihood function or some monotone function of it.

The following is a formalism that is rich enough to embrace most of the "signal-in-noise" problems encountered in signal and image processing. Let $\{x_k\}_{-\infty}^{\infty}$ denote a process with state variable representation

$$x_{k+1}=f_k(x_k,u_k)$$
$$y_k=g_k(x_k). \tag{1}$$

Here f_k and g_k may be random functions; the sequence $\{u_k\}$ is a parameter, decision, or control sequence that may be functionally dependent on the measurement sequence $\{y_k\}$. The range spaces for the state x_k, the parameter u_k, and the measurement y_k are X, U, Y, respectively. These

Manuscript received April 13, 1981. This work was supported in part by the Army Research Office, Research Triangle Park, NC, under Contract DAAG29-79-C-0176 and by the Office of Naval Research, Statistics and Probability Branch, Arlington, VA under Contract N00014-75-C-0518. A preliminary version of this work was presented in an invited session on Signal Processing and Control Interactions at the Thirteenth Annual Asilomar Conference on Circuits, Systems and Computers, Asilomar, CA, November 5–7, 1979.

The authors are with the Department of Electrical Engineering, Colorado State University, Fort Collins, CO 80523.

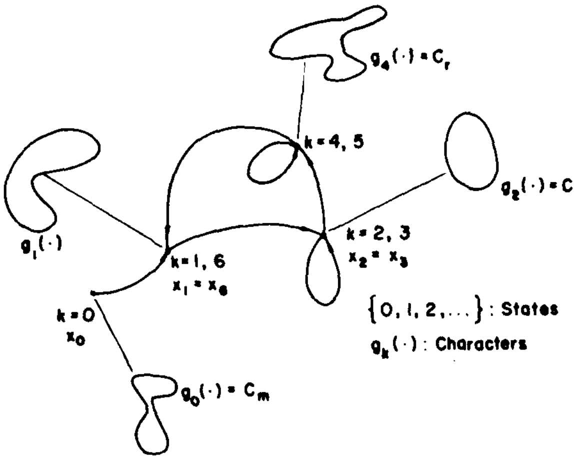


Fig. 1. States and characters.

spaces may be finite, countable, or noncountable. When the spaces X and U are countable then their respective elements may be placed in one-to-one correspondence with the integers and the formalism of Markov chain theory may be mined. Even though the states of X may be chosen abstractly and appear uninteresting, the mapping g_k may be chosen so that the signal component of $g_k(\cdot)$ generates characters or observations C_k that are of great interest. The idea is simply to let a Markov chain, say on the integers $0,1,2,\cdots$, control the dynamical state of the problem and reserve the role of character or observation generation for the observation mechanism $g_k(\cdot)$. This point is illustrated in Fig. 1 where the generated characters can be almost anything: contours, sequences, images, etc.

Consider a finite version of the process $\{x_k\}_{-\infty}^{\infty}$:

$$\begin{aligned} X_N &= (x_0, x_1, \cdots, x_N) \\ &= F_N(X_{N-1}, U_{N-1}) \\ U_N &= (u_0, \cdots, u_N) \\ Y_N &= (y_0, y_1, \cdots, y_N). \end{aligned} \tag{2}$$

Typically, one wants to maximize a performance criterion

$$l_N(X_N, U_N, Y_N) \tag{3}$$

with respect to U_N, subject to constraints $C_N(X_N, U_N)=0$. Call $l_N^N(X_N, U_N, Y_N)$ the maximum. When l_N^N obeys a recursion of the form

$$\begin{aligned} l_N^N(X_N, U_N^N, Y_N) = l_{n-1}^{N-1}(&X_{N-1}, U_{N-1}^{N-1}, Y^N) \\ &+ P_N(x_N, u_N^N, y_N) \end{aligned} \tag{4}$$

then dynamic programming comes to the fore and the solution U_N^N may be generated recursively as the limit of the following sequence of solutions:

$$U_n^n = S_n(U_{n-1}^{n-1}, Y^n), \qquad n=1,2,\cdots,N. \tag{5}$$

The functional S_n describes the recursion for computing U_n^n. Thus the central theme is to imbed the solution to an N stage problem in a sequence of simpler n stage problems. When the underlying state and parameter spaces are finite, the solution algorithm is finite-dimensional and implementable on a digital computer. When they are uncountable, but the function l is quadratic, then it is still often possible to find a closed-form recursive solution that may be programmed.

A very large class of problems may be formulated as before. Two particularly noteworthy examples are the linear discrete-time quadratic regulator problem in deterministic and stochastic control, and Markov chain sequence estimation in additive noise. On the other hand, there are a great number of problems that admit dynamic programming solutions, but which are not naturally formulated in the style above.

One of the points we wish to make is the following: recognizing that a solution is a limit of a sequence of approximants which may be recursively computed is perhaps more fundamental than the search for a corresponding optimization problem. The chief value of an optimization formulation is that it often simplifies the search for the recursive solution algorithm.

III. Dynamic Programming, the DFT, and the FFT

The DFT certainly constitutes one of the cornerstones of modern Fourier analysis. Its uses range over the entire spectrum (so to speak) of signal processing applications. The DFT is a mapping, DFT: $\{x_n\}_0^{N-1} \rightarrow \{X_m\}_0^{N-1}$, that takes the sequence $\{x_n\}_0^{N-1}$ into the sequence $\{X_m\}_0^{N-1}$ according to the rule

$$\begin{aligned} X_m &= \sum_{k=0}^{N-1} x_n W_N^{mn}, \qquad m=0,1,\cdots,N-1 \\ W_N &= \exp(-j2\pi/N). \end{aligned} \tag{6}$$

Noting that $W_N^{-mN}=1$, $\forall m$, we may write X_m as follows:

$$X_m = \sum_{n=0}^{N-1} x_n W_N^{-m(N-n)}. \tag{7}$$

This calculation may be viewed as the limit of the following sequence of imbedded approximations:

$$X_m^{(k)} = \sum_{n=0}^{k-1} x_n W_N^{-m(k-n)}, \qquad k=1,2,\cdots,N. \tag{8}$$

Note $X_m^{(k)}$ = obeys the following recursion:

$$\begin{aligned} X_m^{(k+1)} &= W_N^{-m} X_m^{(k)} + W_N^{-m} x_k \\ X_m^{(N)} &= X_m \\ X_m^{(1)} &= x_0 W_N^{-m}. \end{aligned} \tag{9}$$

So X_m is obtained as the limit of a sequence of approximations that begins at $X_m^{(1)} = x_0 W_N^{-m}$ and terminates at $X_m^{(N)} = X_m$. This is the so-called Goertzel algorithm [8] for obtaining the mth DFT variable X_m, as the output of a digital filter excited by the sequence $\{x_n\}_0^{N-1}$. The output of the filter is read at time $k=N$ (see Fig. 2).

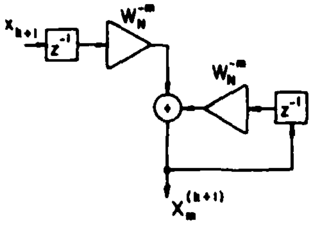


Fig. 2. Goertzel filter for DFT component X_m.

Dynamic Programming and the Decimation-in-Frequency FFT

The Goertzel algorithm is a nice dynamic programming-like solution for the DFT. However, it is not efficient. Computational complexity is of order N^2. Let us see if we can improve upon it. Consider $X_m^{(k)}$ for even frequency indices $m=2r$:

$$\begin{aligned} X_{2r}^{(k)} &= \sum_{n=0}^{k-1} x_n W_N^{-2r(k-n)}, \qquad k=1,2,\cdots,N \\ &= \sum_{n=0}^{k-1} x_n W_{N/2}^{-r(k-n)}, \qquad k-1,2,\cdots,N. \end{aligned} \tag{10}$$

For k even (say $k=2s$),

$$\begin{aligned} X_{2r}^{(2s)} &= \sum_{n=0}^{2s-1} x_n W_{N/2}^{-r(2s-n)} \\ &= \sum_{n=0}^{s-1} x_n W_{N/2}^{-r(2s-n)} + \sum_{n=s}^{2s-1} x_n W_{N/2}^{-r(2s-n)} \\ &= W_{N/2}^{-rs} \sum_{n=0}^{s-1} x_n W_{N/2}^{-r(s-n)} + \sum_{l=0}^{s-1} x_{l+s} W_{N/2}^{-r(s-l)} \\ &= W_{N/2}^{-rs} X_{2r}^{(s)} + Y_{2r}^{(s)}. \end{aligned} \tag{11}$$

This shows that the two s-point DFT approximant $X_{2r}^{(2s)}$ may be obtained from two s-point approximants. By choosing $s=N/2$ and continuing backwards in this way (for odd subindices, as well) one arrives at a backward dynamic programming derivation of the decimation-in-frequency FFT See Fig. 3 for an elementary representation of a four-point decimation in frequency FFT. The decimation-in-frequency algorithm improves on the Goertzel algorithm by requiring complexity on order $N\log N$.

IV. Detection, Estimation, and Control in the AR(N) Case: Kalman Filters, Levinson Recursions, and Dynamic Programming

Autoregressive (AR) models for signals, states, and data play a starring role in many areas of signal processing and control. By appropriately selecting model parameters (and order) one can model the covariance structure and spectral characteristics of more general models. The so-called normal equations for identifying AR parameters are elegant and easily solved with recursions of the Levinson type.

In this section we tie up control, prediction, detection, and estimation in the special case where we are dealing with a zero-mean wide-sense stationary, scalar autoregressive time series. The usual state-variable and matrix block diagrams give way to scalar variables and digital filter blocks of moving average filters. The normal equations are highlighted and dynamic programming is used to derive the famous Levinson recursions.

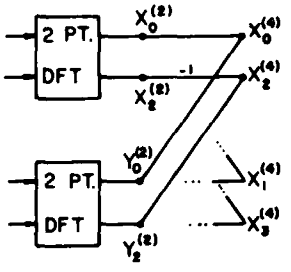


Fig. 3. Four-point decimation-in-frequency FFT.

A. Models

Let $\{x_k\}$ denote a scalar zero-mean wide-sense stationary AR sequence of order p (denoted AR(p)) that obeys the recursion

$$x_k = \sum_{n=1}^{p} a_n x_{k-n} + w_k, \qquad \forall k.$$

$$w_n\text{: sequence of i.i.d. } N(0,\sigma_w^2) \text{ random variables (r.v.s.).}^1 \tag{12}$$

It is easy to see that the covariance sequence $\{r_m\}_{-\infty}^{\infty}$, $r_m = r_{-m}$, associated with the sequence $\{x_k\}$ obeys the recursion

$$r_m = \sum_{n=1}^{p} a_n r_{m-n} + \sigma_w^2 \delta_m, \qquad m=0,1,\cdots. \tag{13}$$

From here one may write out the so-called normal equations:

$$\begin{aligned} R_p a_p &= r_p \\ R_p &= \begin{bmatrix} r_0 & r_1 & \cdots & r_{p-1} \\ r_1 & r_0 & & r_{p-2} \\ \vdots & & \ddots & \vdots \\ r_{p-1} & & \cdots & r_0 \end{bmatrix} \\ a_p' &= (a_1,\cdots,a_p) \\ r_p' &= (r_1,\cdots,r_p). \end{aligned} \tag{14}$$

We note at this juncture that turning r_p upside down turns the solution to the normal equations upside down. To see this, let

$$J = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 0 & & & 1 & 0 \\ \vdots & & \cdot^{\cdot^{\cdot}} & & \vdots \\ & & & & 0 \\ 1 & & \cdots & 0 & 0 \end{bmatrix}, \qquad JJ=I \tag{15}$$

denote the exchange matrix and note by the Toeplitz symmetry of R_p we have $JR_pJ = R_p$. Thus

[1] Here and elsewhere i.i.d. stands for independent, identically distributed and r.v. stands for random variable.

$$JR_pJ\mathbf{a}_p=\mathbf{r}_p$$
$$R_pJ\mathbf{a}_p=J\mathbf{r}_p \tag{16}$$

As J turns vectors upside down, this proves the claim.

B. The Normal Equations are Fundamental

The AR coefficients $\mathbf{a}_p'=(a_1,\cdots,a_p)$ that characterize the sequence $\{x_k\}$ are fundamental to the implementation of control, prediction, and detection algorithms on noisily observed AR sequences. Unfortunately, sequences rarely come tagged with their corresponding AR parameters. More typically finite records of them come to use and we estimate a covariance function (or power spectrum), often by FFT-ing, squaring and windowing, and inverse FFT-ing. These estimates may then be used to solve for the coefficients $\mathbf{a}_p$ from the normal equations. This makes the normal equations fundamental and arouses our interest in efficient ways of solving them. The derivation that follows is an adaptation of Bellman's discussion of quadratic forms and dynamic programming in [9].

C. Dynamic Programming and Levinson's Algorithm

Consider the quadratic form

$$Q_p(\mathbf{r}_p)=r_0-2\boldsymbol{\alpha}_p'\mathbf{r}_p+\boldsymbol{\alpha}_p'R_p\boldsymbol{\alpha}_p \tag{17}$$

This quadratic form is minimized for some choice of $\boldsymbol{\alpha}_p$ that we denote $\boldsymbol{\alpha}_p^p$. It is easy to see that

$$\boldsymbol{\alpha}_p^p=\mathbf{a}_p \tag{18}$$

where $\mathbf{a}_p$ comes from the normal equation:

$$\mathbf{a}_p=R_p^{-1}\mathbf{r}_p. \tag{19}$$

The corresponding minimum of $Q_p(\mathbf{r}_p)$ we denote $Q_p^p(\mathbf{r}_p)$:

$$\begin{aligned}Q_p^p(\mathbf{r}_p)&=r_0-\boldsymbol{\alpha}_p^{p\prime}\mathbf{r}_p\\&=r_0-\mathbf{r}_p'R_p^{-1}\mathbf{r}_p.\end{aligned} \tag{20}$$

The quadratic form $Q_p(\cdot)$ may be written recursively as

$$Q_p(\mathbf{r}_p)=\alpha_p^2r_0-2\alpha_pr_p+Q_{p-1}(\mathbf{r}_{p-1}-\alpha_pJ\mathbf{r}_{p-1}). \tag{21}$$

So minimization of $Q_p(\mathbf{r}_p)$ with respect to $\boldsymbol{\alpha}_p$ may be written

$$\begin{aligned}Q_p^p(\mathbf{r}_p)&=\min_{\boldsymbol{\alpha}_p}Q_p(\mathbf{r}_p)\\&=\min_{\alpha_p}\left[\alpha_p^2r_0-2\alpha_pr_p+\min_{\boldsymbol{\alpha}_{p-1}}Q_{p-1}(\mathbf{r}_{p-1}-\alpha_pJ\mathbf{r}_{p-1})\right]\\&=\min_{\alpha_p}\left[\alpha_p^2r_0-2\alpha_pr_p+Q_{p-1}^{p-1}(\mathbf{r}_{p-1}-\alpha_pJ\mathbf{r}_{p-1})\right].\end{aligned} \tag{22}$$

This equation contains the essence of dynamic programming and the principle of optimality: once the solution $\boldsymbol{\alpha}_{p-1}^{p-1}$, and corresponding minimum Q_{p-1}^{p-1}, have been found for the order $(p-1)$ problem, α_p^p may be found as a function of r_0, $\mathbf{r}_p$, and $\boldsymbol{\alpha}_{p-1}^{p-1}$. At each step of the way the minimization on $\boldsymbol{\alpha}_p$ is quadratic.

Use (20) in (22) to get a different recursion for $Q_p^p(\mathbf{r}_p)$:

$$\begin{aligned}Q_p^p(\mathbf{r}_p)&=\min\left[\alpha_p^2r_0-2\alpha_pr_p+r_0-(\mathbf{r}_{p-1}-\alpha_pJ\mathbf{r}_{p-1})'\right.\\&\qquad\left.\cdot R_{p-1}^{-1}(\mathbf{r}_{p-1}-\alpha_pJ\mathbf{r}_{p-1})\right]\\&=\min_{\alpha_p}\left[\alpha_p^2(r_0-\mathbf{r}_{p-1}'JR_{p-1}^{-1}J\mathbf{r}_{p-1})\right.\\&\qquad\left.-2\alpha_p(r_p-\mathbf{r}_{p-1}'R_{p-1}^{-1}J\mathbf{r}_{p-1}+Q_{p-1}^{p-1}(\mathbf{r}_{p-1})\right]\\&=\min_{\alpha_p}\left[\alpha_p^2(r_0-\boldsymbol{\alpha}_{p-1}^{p-1\prime}\mathbf{r}_{p-1})\right.\\&\qquad\left.-2\alpha_p(r_p-\boldsymbol{\alpha}_{p-1}^{p-1\prime}J\mathbf{r}_{p-1})+Q_{p-1}^{p-1}(\mathbf{r}_{p-1})\right].\end{aligned} \tag{23}$$

It follows easily that the minimizing value of α_p is

$$\alpha_p^p=\frac{r_p-\boldsymbol{\alpha}_{p-1}^{p-1\prime}J\mathbf{r}_{p-1}}{r_0-\boldsymbol{\alpha}_{p-1}^{p-1\prime}\mathbf{r}_{p-1}}. \tag{24}$$

Equivalently,

$$(\alpha_p^p)^2(r_0-\boldsymbol{\alpha}_{p-1}^{p-1\prime}\mathbf{r}_{p-1})=\alpha_p^p(r_p-\boldsymbol{\alpha}_{p-1}^{p-1\prime}J\mathbf{r}_{p-1}). \tag{25}$$

Substituting the solution of (24) into (23) we get the following recursion for $Q_p^p(\mathbf{r}_p)$:

$$\begin{aligned}Q_p^p(\mathbf{r}_p)&=Q_{p-1}^{p-1}(\mathbf{r}_{p-1})-\alpha_p^p(r_p-\boldsymbol{\alpha}_{p-1}^{p-1\prime}J\mathbf{r}_{p-1}\\&=Q_{p-1}^{p-1}(\mathbf{r}_{p-1})-(\alpha_p^p)^2(r_0-\boldsymbol{\alpha}_{p-1}^{p-1\prime}\mathbf{r}_{p-1}).\end{aligned} \tag{26}$$

Comparing this with (20), we have

$$\boldsymbol{\alpha}_p^{p\prime}\mathbf{r}_p=\boldsymbol{\alpha}_{p-1}^{p-1\prime}\mathbf{r}_{p-1}+\alpha_p^p(r_p-\boldsymbol{\alpha}_{p-1}^{p-1\prime}J\mathbf{r}_{p-1}) \tag{27}$$

from whence, it follows that

$$\boldsymbol{\alpha}_p^{p\prime}=(\boldsymbol{\alpha}_{p-1}^{p-1\prime}-\alpha_p^p\boldsymbol{\alpha}_{p-1}^{p-1\prime}J,\alpha_p^p). \tag{28}$$

This is the recursion for updating $\boldsymbol{\alpha}_p^p$.

This completes the Levinson recursions, summarized as follows:

$$\begin{aligned}&\text{(i)}\quad \alpha_p^p=\frac{r_p-\boldsymbol{\alpha}_{p-1}^{p-1\prime}J\mathbf{r}_{p-1}}{r_0-\boldsymbol{\alpha}_{p-1}^{p-1\prime}\mathbf{r}_{p-1}}\\&\text{(ii)}\quad \boldsymbol{\alpha}_p^{p\prime}=(\boldsymbol{\alpha}_{p-1}^{p-1\prime}-\alpha_p^p\boldsymbol{\alpha}_{p-1}^{p-1\prime}J,\alpha_p^p)\\&\text{(iii)}\quad Q_p^p(\mathbf{r}_p)=Q_{p-1}^{p-1}(\mathbf{r}_{p-1})-(\alpha_p^p)^2(r_0-\boldsymbol{\alpha}_{p-1}^{p-1\prime}\mathbf{r}_{p-1}).\end{aligned} \tag{29}$$

There are no matrix inverses here—only vector inner products. Thus the algorithm is $O(p^2)$. Of course, $\boldsymbol{\alpha}_p^p$ is the desired solution $\mathbf{a}_p$.

To show the importance of the AR coefficients $\mathbf{a}_p=\boldsymbol{\alpha}_p^p$, we consider the following family of problems:

(i) noisy prediction
(ii) noise-free prediction
(iii) minimum variance control
(iv) detection.

D. Noisy Prediction and the Kalman Predictor

Assume the sequence $\{x_k\}$ is observed in zero-mean additive white Gaussian noise (WGN):

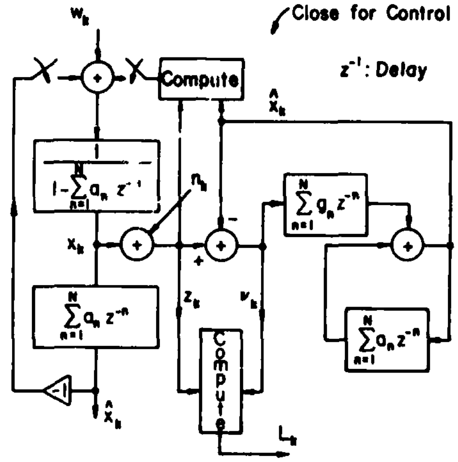

w_k
Close for Control
z^{-1}: Delay
Compute
$\hat{x}_k$
$\frac{1}{1-\sum_{n=1}^{N} a_n z^{-1}}$
n_k
$\sum_{n=1}^{N} g_n z^{-n}$
x_k
−
+
$\sum_{n=1}^{N} a_n z^{-n}$
z_k
ν_k
$\sum_{n=1}^{N} a_n z^{-n}$
Compute
−1
$\hat{x}_k$
L_k

Fig. 4. Prediction, detection, and control of a noisy AR(N) sequence.

$$z_k = x_k + n_k, \qquad \forall k$$

$$n_k\text{: sequence of i.i.d. } N(0,\sigma_n^2) \text{ random variables.} \tag{30}$$

The companion form state model is

$$\begin{aligned} X_{k+1} &= AX_k + bw_{k+1} \\ Z_k &= c'X_k \\ X_k &= \begin{bmatrix} x_{k-p+1} \\ \vdots \\ x_{k-1} \\ x_k \end{bmatrix} \\ A &= \left[\begin{array}{cccc} 0 & & & \\ & & I & \\ 0 & & & \\ \hline a_p & a_{p-1} & \cdots & a_0 \end{array}\right] \\ b = c &= \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}. \end{aligned} \tag{31}$$

The stationary Kalman one-step predictor for the noisily observed AR(p) sequence is

$$\begin{aligned} \hat{X}_{k+1} &= A\hat{X}_k + K(z_k - \hat{x}_k) \\ \hat{x}_k &= c'\hat{X}_k \end{aligned} \tag{32}$$

where K is a ($p \times 1$) Kalman gain:

$$\begin{aligned} K &= APc\left(c'Pc + \sigma_n^2\right)^{-1} \\ &= [k_1, k_2, \cdots, k_N]' \\ P &= (A - Kc')P(A - Kc')' + \sigma_n^2 KK' + \sigma_w^2 bb'. \end{aligned} \tag{33}$$

The Kalman prediction sequence $\{\hat{x}_{k+1}\}$ for $\{z_{k+1}\}$ can be interpreted as the output of an AR moving average (ARMA) filter with p poles and $p-1$ zeros (denoted ARMA($p, p-1$)), driven by the prediction error sequence $\{\nu_k = z_k - x_k\}$ or an ARMA($p, p-1$) filter driven by the observation sequence $\{z_k\}$. The resulting filter equations are

$$\hat{x}_{k+1} = \sum_{i=1}^{p} a_i \hat{x}_{k+1-i} + \sum_{i=1}^{p} g_i \nu_{k+1-i} \tag{34}$$

or

$$\hat{x}_{k+1} = \sum_{i=1}^{p} (a_i - g_i)\hat{x}_{k+1-i} + \sum_{i=1}^{p} g_i z_{k+1-i} \tag{35}$$

where the coefficients g_i can be defined by the characteristic polynomial $\Delta(\lambda)$ of ($A - Kc'$):

$$\Delta(\lambda) = \lambda^p + \sum_{i=1}^{p} (g_i - a_i)\lambda^{p-i}. \tag{36}$$

See Fig. 4 for a block diagram of this predictor. Note that the noise-free moving average (MA) predictor filter, $P(z) = \sum_{n=1}^{p} a_n z^{-n}$, is preserved in the feedback loop, but that the residual sequence $\nu_k = z_k - \hat{x}_k$ is now weighted with a feedforward MA filter, $Q(z) = \sum_{n=1}^{p} g_n z^{-n}$.

Why is the noisy Kalman predictor ARMA and not MA? The answer is that $\{z_k\}$, a noisy version of an AR signal process, obeys an ARMA(p, p) difference equation. As an AR(p) model has an MA($p-1$) predictor, it is at least logical (if not intuitive) that an ARMA(p, p) process has an ARMA($p, p-1$) predictor.

E. The Noise-Free Predictor

The prediction vector $\hat{X}_k$ consists of the terms

$$\begin{matrix} E[x_{k-p+1}/z_{k-1}, z_{k-2}, \cdots] \\ \vdots \\ E[x_{k-1}/z_{k-1}, z_{k-2}, \cdots] \\ E[x_k/z_{k-1}, z_{k-1}, \cdots]. \end{matrix} \tag{37}$$

When $\sigma_n^2 = 0$, then $z_k = x_k, \forall k$ and

$$E[x_{k-n}/z_{k-n}, z_{k-n-1}, \cdots] = E[x_{k-n}/x_{k-n}, \cdots] = x_{k-n}, \qquad n = 1, 2, \cdots. \tag{38}$$

So in this case the prediction vector is

$$\hat{X}_k = \begin{bmatrix} x_{k-p+1} \\ x_{k-p+2} \\ \vdots \\ x_{k-1} \\ \hat{x}_{k/k-1} \end{bmatrix}. \tag{39}$$

It follows that P, the covariance $E[X_k - \hat{X}_k][X_k - \hat{X}_k]'$ is

$$P = \begin{bmatrix} 0 & \cdots & & 0 \\ \vdots & & & \vdots \\ & & & 0 \\ 0 & \cdots & 0 & \sigma_w^2 \end{bmatrix}. \tag{40}$$

Calculating K by substituting (40) into (33), we find that $\Delta(\lambda) = \lambda^N$ and hence $g_i = a_i$. This implies, as one would expect, that the prediction filter of (35) reduces to the purely MA relation:

$$\hat{x}_{k+1} = \sum_{i=1}^{p} a_i z_{k+1-i}. \tag{41}$$

F. Minimum Variance Control

One of the simplest control strategies is minimum variance regulation where one desires to minimize the variance of the AR(p) output sequence $\{x_k\}$, and force $E(x_k)=0$. The well known separation principle allows one to generate a feedback control strategy assuming noise free measurements, i.e., $n_k = 0$, and then use the same strategy in the noisy case but with the Kalman filter estimates $\{\hat{x}_k\}$ replacing the actual filter outputs $\{x_k\}$.

Assume then we have the system

$$x_k = \sum_{i=1}^{p} a_i x_{k-i} + w_k + v_k \tag{42}$$

where $\{v_k\}$ is our feedback control sequence. We would like to minimize

$$\begin{aligned} E(x_k^2) &= E\left(\sum_{i=1}^{p} a_i x_{k-i} + w_k + v_k\right)^2 \\ &= E\left(w_k^2 + 2w_k v_k + 2w_k\left(\sum_{i=1}^{p} a_i x_{k-i}\right)\right. \\ &\quad \left. + \left(v_k + \sum_{i=1}^{p} a_i x_{k-i}\right)^2\right) \\ &= E(w_k^2) + 2E(E(w_k v_k / v_k) v_k) \\ &\quad + 2E\left(w_k\left(\sum_{i=1}^{p} a_i x_{k-i}\right)\right) \\ &\quad + E\left(\left(v_k + \sum_{i=1}^{p} a_i x_{k-i}\right)^2\right). \end{aligned} \tag{43}$$

Since $\{w_k\}$ is uncorrelated with $\{x_{k-i}\}$, $i \geqslant 1$, and since $E(w_k v_k | v_k) = 0$, it is clear that $E(x_k^2)$ is minimized by choosing

$$v_k - - \sum_{i=1}^{p} a_i x_{k-i}. \tag{44}$$

This control is illustrated in Fig. 4 as a feedback loop running up the left side of the figure. The feedback loop to the top "compute" box shows how $\hat{x}_k$ would be used for minimum variance control in the noisy case.

G. Det ction and the Likelihood Ratio

Consider the hypothesis test H_0 versus H_1 with

$$\begin{aligned} &H_0: z_k = n_k, \qquad k = 0, 1, \cdots, K \\ &H_1: z_k = x_k + n_k, \qquad k = 0, 1, \cdots, K \end{aligned} \tag{45}$$

and the data assumed stationary over the interval. This test is equivalent to the test $\hat{H}_0$ versus $\hat{H}_1$, where

$$\begin{aligned} &\hat{H}_0: \nu_k^{(0)} = z_k: N(0, \sigma_n^2) \\ &\hat{H}_1: \nu_k^{(1)}: N(0, p_0 + \sigma_n^2); p_0: \text{variance of } \hat{x}_k \end{aligned} \tag{46}$$

and $\nu_k^{(1)} = z_k - x_k$ is the innovations sequence in the Kalman filter. The log-likelihood ratio for this problem is proportional to

$$\mathrm{LR} = K - \frac{1}{P_0 + \sigma_n^2} \sum_{k=0}^{K} \nu_k^2 + \frac{1}{\sigma_n^2} \sum_{k=0}^{K} z_k^2. \tag{47}$$

Thus the statistics $\Sigma \nu_k^2$ and Σz_k^2 are sufficient and the log-likelihood ratio may be computed as in Fig. 4.

V. Frequency Tracking and Dynamic Programming

Phase and frequency tracking problems comprise some of the most nettlesome nonlinear filtering problems in the entire realm of signal processing. Nonlinear filtering and MAP solutions have been reported recently by Bucy and Mallinckrodt [11], Ungerboeck [12], Tufts [13], Scharf *et al.* [14], [15], and Wolcin [16]. A typical problem is the following: observe the signal-plus-noise sequence $\{z_k\}$ with

$$\begin{aligned} &z_k = s_k + n_k;\ n_k: \text{sequence of i.i.d. } N(0, \sigma_n^2) \\ &\qquad \text{random variables.} \\ &s_k = e^{j\phi_k} \end{aligned} \tag{48}$$

and estimate the phase sequence $\{\phi_k\}$ or some underlying function of it. Here the character assigned to state ϕ_k is $s_k = \exp(j\phi_k)$.

In all that follows it will be convenient to organize the observed data into contiguous data blocks:

$$Z = \begin{bmatrix} z_0 \\ z_1 \\ \vdots \\ z_{K-1} \end{bmatrix} \qquad z_i - s_i + n_i$$

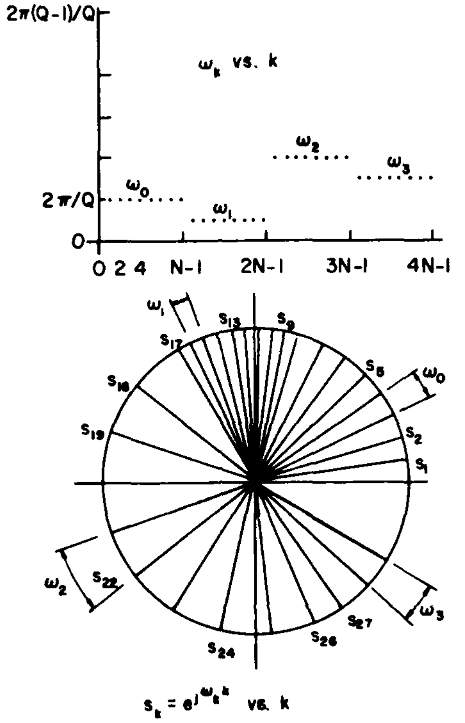


Fig. 5. Visualizing the random walk frequency trajectories.

$$s_t=\begin{bmatrix} s_{tN} \\ \vdots \\ s_{(t+1)N-1}\end{bmatrix} \qquad n_t=\begin{bmatrix} n_{tN} \\ \vdots \\ n_{(t+1)N-1}\end{bmatrix} \tag{49}$$

Think of the $(NK\times 1)$ vector Z as a concatenation of K data blocks of the form z_t, each of dimension N.

The choice of a model for ϕ_k determines whether we are talking about phase or frequency tracking, although the distinction between the two is more imagined than real. In [14] the phase was assumed to evolve according to a random walk phase model

$$\phi_k=\phi_{k-1}+w_k$$

$$w_k\text{: sequence of i.i.d. } N(0,\sigma_w^2)\text{ random variables} \tag{50}$$

and a complete dynamic programming solution was presented. In [15] a discontinuous-phase FM model was assumed and a dynamic program algorithm derived.

Here we assume ϕ_k evolves according to the continuous-phase FM rule

$$\phi_k=\phi_{k-1}+\omega_k \tag{51}$$

where[2]

$$\begin{aligned}\omega_k&=\frac{2\pi}{Q}\nu_{[k/N]}\\ &=\frac{2\pi}{Q}\nu_t, \qquad tN\leqslant k<(t+1)N.\end{aligned} \tag{52}$$

In this way ω_k is fixed at the value $(2\pi/Q)\nu_t$ for a block of N as illustrated in Fig. 5. Correspondingly, ϕ_k increases at a fixed linear rate for N samples and then adopts a new linear rate for the next N samples. Therefore, we may write s_t as follows:

$$s_t=\exp(j\phi_{tN})d(\nu_t)$$

$$d(\nu_t)=\begin{bmatrix}\exp\left(j\frac{2\pi}{Q}\nu_t\right)\\ \vdots \\ \exp\left(j\frac{2\pi}{Q}(N-1)\nu_t\right)\end{bmatrix}. \tag{53}$$

The phase ϕ_{tN} is the total accumulated phase after tN steps. It depends on the entire history of frequency terms $\nu_0,\nu_1,\cdots,\nu_{t-1}$ and obeys the recursion

$$\phi_{tN}=\phi_{(t-1)N}+N\frac{2\pi}{Q}\nu_{t-1}.$$

Between tN and $(t+1)N-1$ the phase grows linearly as

$$\begin{aligned}\phi_i&=\phi_{tN}+(i-tN)\frac{2\pi}{Q}\nu_t\\ &=\phi_{[i]N}+(i-[i]N)\frac{2\pi}{Q}\nu_t.\end{aligned}$$

This additional phase increase is accounted for in the vector $d(\nu_t)$. See Fig. 5 for an illustration.

To complete the model we assume $\{\nu_t\}$ is a sequence of discrete random variables that take values in the set $\{0,1,\cdots,Q-1\}$ and evolve according to the rule

$$\nu_t=\nu_{t-1}\oplus u_t \tag{54}$$

where $u_t\in\{0,1,\cdots,Q-1\}$ and addition is modulo-Q. The distribution of the sequence of i.i.d. random variables is selected in such a way that the transition probability

$$p(\nu_t/\nu_{t-1}) \tag{55}$$

corresponds to our notion of physical reality. We may think of the resulting frequency sequence $\{\omega_k\}$ as a finite-state random walk on the circle with an unusual transition probability structure. Typical trajectories for $\{\omega_k\}$ and $\{s_k\}$ are illustrated in Fig. 5.

The joint likelihood function for Z and $\{\nu_t\}$ is proportional to

$$l=\sum_{t=0}^{K-1}\frac{1}{2\sigma_n^2}|z_t-s_t|^2+\sum_{t=0}^{K-1}\ln p\left(\frac{\nu_t}{\nu_{t-1}}\right). \tag{56}$$

Using our representation for s_t and dropping terms independent of ν_t we obtain

$$l\sim\frac{1}{2\sigma_n^2}\sum_{t=0}^{K-1}\text{Re}\{\exp(-j\phi_{tN})z_t'd^*(\nu_t)\}+\sum_{t=0}^{K-1}\ln p\left(\frac{\nu_t}{\nu_{t-1}}\right). \tag{57}$$

The term $z_t'd^*(\nu_t)$ is nothing more than the DFT of z_t evaluated at the DFT frequency $(2\pi/Q)\nu_t$. The best way to compute it is to zero-pad z_t to obtain a Q point sequence that may be FFT'ed. See Fig. 6.

Our notion of the most likely sequence $\{\nu_t\}_0^{K-1}$ is the

[2] $[\cdot]$ denotes integer part of $(\cdot)$.

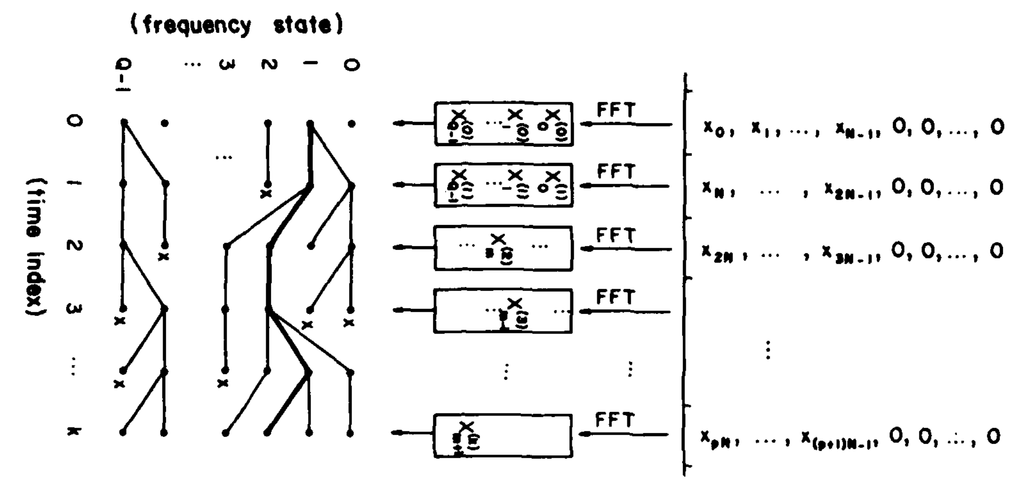

(frequency state)
0
1
2
3
...
Q-1
(time index)
0
1
2
3
...
K
FFT
FFT
FFT
FFT
FFT
x₀, x₁, ..., x_{N-1}, 0, 0, ..., 0
x_N, ..., x_{2N-1}, 0, 0, ..., 0
x_{2N}, ..., x_{3N-1}, 0, 0, ..., 0
x_{pN}, ..., x_{(p+1)N-1}, 0, 0, ..., 0

Fig. 6. Data processing and frequency trellis illustrating evolution of surviving frequency tracks.

sequence that maximizes l. This is the MAP sequence. Write the maximization problem as

$$\max_{\{\nu_t\}_0^{K-1}} \Gamma_{K-l} \tag{58}$$

with

$$\Gamma_t = \Gamma_{t-1} + \frac{1}{\sigma_n^2}\operatorname{Re}\{\exp(-j\phi_{tN}) z_t' d^*(\nu_t)\} + \ln p\left(\frac{\nu_t}{\nu_{t-1}}\right). \tag{59}$$

So our maximization problem becomes

$$\max_{\{\nu_t\}_{K-2}^{K-1}} \left[\max_{\{\nu_t\}_0^{K-3}} \Gamma_{K-2} + \ln p\left(\frac{\nu_{K-1}}{\nu_{K-2}}\right) + \frac{1}{2\sigma_n^2}\operatorname{Re}\exp(-j\phi_{(K-1)N}) z_{(K-1)}' d^*(\nu_{(K-1)}) \right]. \tag{60}$$

Thus for each node on Fig. 6 we evaluate the FFT $z_t' d^*(\nu_t)$, phase it by $\exp(-j\phi_{tN})$, and find the best route through the trellis with the dynamic programming algorithm of (61). This completes our algorithm for moderating the usual peak-picking rule on the FFT with prior information $p(\nu_t/\nu_{t-1})$. The reader is referred to [14] for a more complete discussion of a related algorithm for nonlinear phase tracking.

Shown in Figs. 7 and 8 are simulations of the algorithm running on noisy phase-coherent FM data. The parameters are

$$\begin{aligned} &u_t\text{: folded normal r.v. of [14]} \\ &\text{variance}\left(\frac{2\pi}{Q}u_t\right) = \begin{cases} 0.01, & \text{Fig. 7} \\ 0.1, & \text{Fig. 8} \end{cases} \qquad Q = 32 \\ &\text{CNR} = 10\log_{10}\frac{1}{\sigma_n^2} = -3\text{ dB} \end{aligned} \tag{61}$$

where CNR is the carrier-to-noise ratio.

The algorithm is run in a fixed lag mode [14] for a lag of 60. The results show that even when FFT peaks, indicated with 1 (peak), 2 (secondary peak), and 3 (tertiary peak) are unreliable the estimated frequency sequence (thin line) tracks the true frequency sequence (thick line).

This formulation improves on a heuristic idea of Rockmore, who was perhaps the first to advocate dynamic programming search for likely frequency tracks [17].

VI. Local Boundary Estimation in Noisy Black and White Images

In digital image processing one is interested in developing computer algorithms which can either automatically extract information from pictures or at least simplify the process of manually interpreting them. In either case, a basic step involves segmenting a picture into regions with similar features such as gray level or texture. This involves the estimation of region boundaries. Boundary estimation algorithms make use of operators which estimate short segments of boundaries using picture data in small picture sections. Examples are simple gradient operators and the well-known Hueckel operator [18]. An example of a local sequential estimator which is also used for this purpose can be found in [19]. In this section we outline a new dynamic programming algorithm for sequentially estimating short boundary segments. We then briefly discuss an algorithm which pieces together the short segments and present some examples of its use on complete images.

A. Image and Boundary Models

Let a digitized black and white image be represented by a matrix with components g_{ij} corresponding to the gray level value of a picture element (pixel) centered at position (i, j). The value g_{ij} will have two components—a true picture component b_{ij} and a noise component n_{ij} so that $g_{ij} = b_{ij} + n_{ij}$. A picture is assumed to consist of a single region of gray level r_{in} lying in a background of gray level r_{out}, so that b_{ij} can take on either of the two values r_{in} or r_{out}. The noise components n_{ij} are assumed to be independent identically distributed Gaussian random variables with mean zero and variance σ^2, denoted n_{ij}: $N(0, \sigma^2)$.

An edge element is defined as the line segment separat-

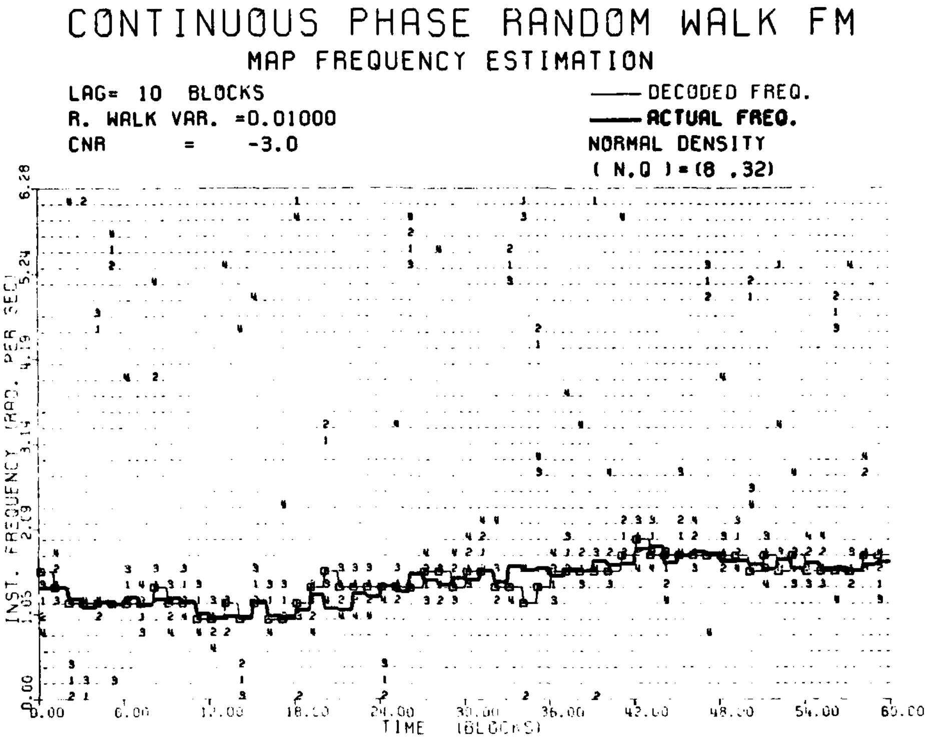


Fig. 7. Frequency tracking at CNR = −3.0 dB. Random walk variance = 0.01 rad^2.

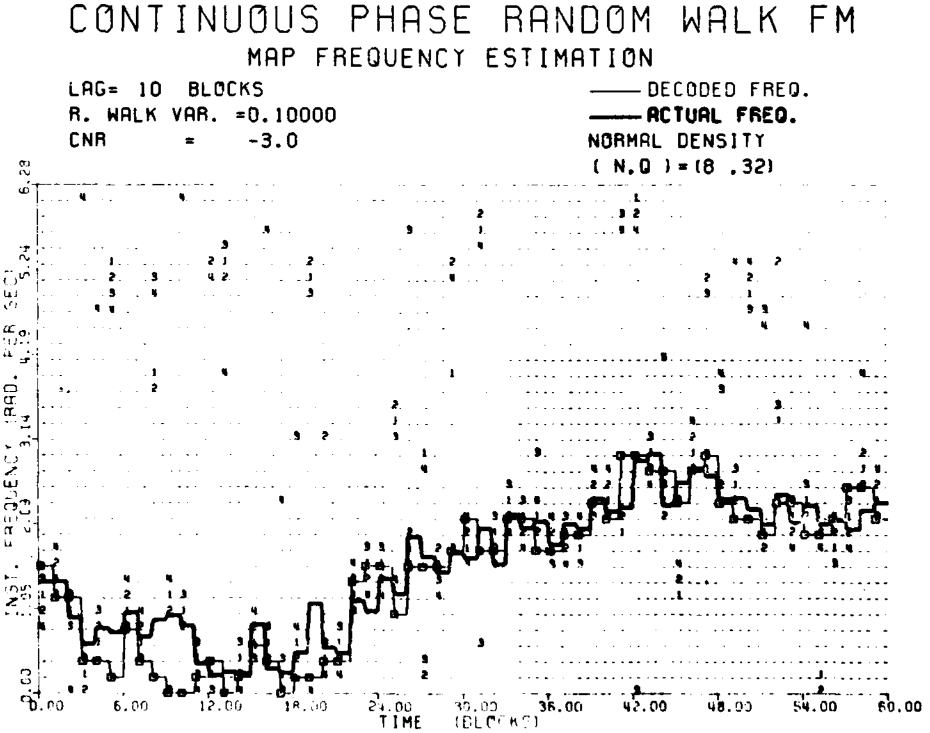


Fig. 8. Frequency tracking at CNR = −3.0 dB. Random walk variance = 0.1 rad^2.

ing two adjacent pixels, and as shown in Fig. 9 a boundary segment consists of a directed sequence of edge elements $\{t_i\}_1^M$. As illustrated in Fig. 10, we assume short boundary segments to be generated by constructing a sequence of edge elements that terminate at the boundary of a rectangular box. Longer sequences of edge elements defining longer more complicated boundaries are obtained by exiting successive rectangles R_k, $1 \le k \le N$ containing $\rho_k = kx(2k-2)$ pixels. The key constraint built into this generating scheme is that sequences departing one rectangle cannot re-enter it. Fig. 11(a) gives an example of a boundary which is consistent with this model while Fig. 11(b) shows

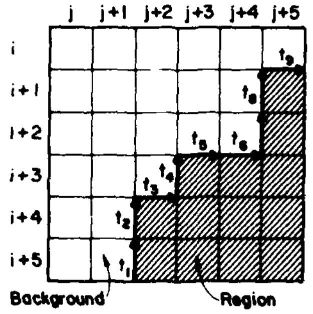


Fig. 9. A boundary segment in small picture segment.

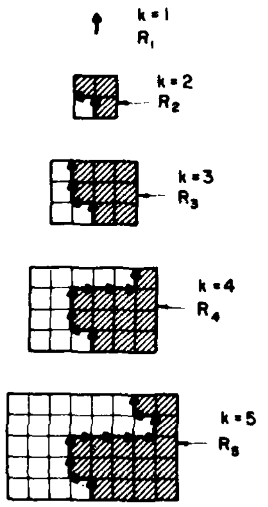


Fig. 10. Example of boundary segment generation.

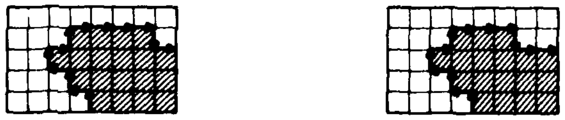


Fig. 11. Example of a boundary. (a) Consistent with model. (b) Inconsistent with model.

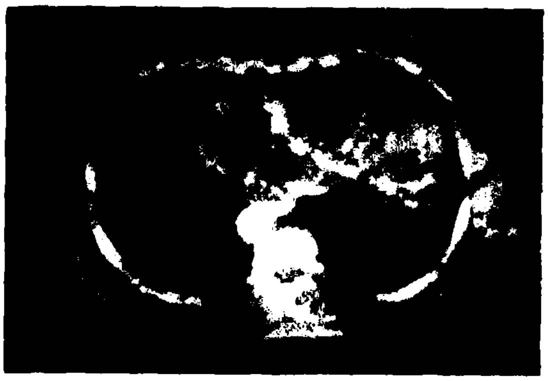


Fig. 12. CAT scan of abdominal section of human body.

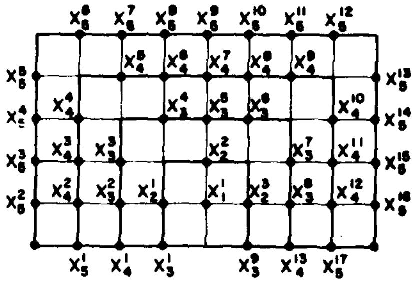

x_5^6 x_5^7 x_5^8 x_5^9 x_5^{10} x_5^{11} x_5^{12}
x_5^5 x_5^4 x_5^3 x_5^2
x_5^{13} x_5^{14} x_5^{15} x_5^{16}
x_4^5 x_4^6 x_4^7 x_4^8 x_4^9
x_4^4 x_3^4 x_3^5 x_3^6 x_4^{10}
x_4^3 x_3^3 x_2^2 x_3^7 x_4^{11}
x_4^2 x_3^2 x_2^1 x_1^1 x_2^3 x_3^8 x_4^{12}
x_5^1 x_4^1 x_3^1 x_3^9 x_4^{13} x_5^{17}

Fig. 13. Possible state locations x_k^j, for $k=1,2,\cdots,5$.

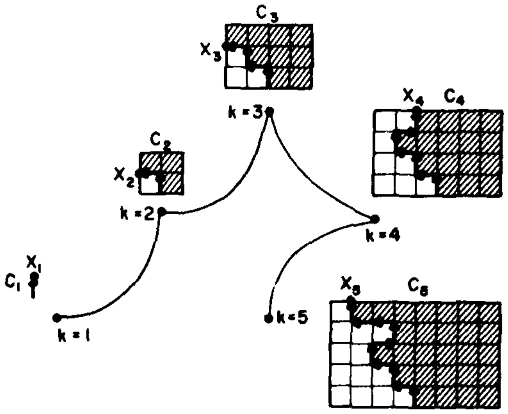


Fig. 14. State transition diagram for $k=1,2,\cdots,5$ illustrating a set of Characters C_k, $k=1,2,\cdots,5$ for a specific process realization x_k, $k=1,2,\cdots,5$.

a similar but inconsistent boundary. In the latter case the edge sequence reenters R_4. Although this scheme restricts somewhat the types of boundary segments that can be generated, it is still very reasonable for region boundaries with low and slowly varying curvatures such as those in the body computerized axial tomography (CAT) scan shown in Fig. 12. The maximum rectangle size ρ_N is assumed fixed *a priori* and is a function of the boundary curvature properties for the region of interest.

Boundary segments generated by such a model are naturally represented by a sequence of states in a Markov chain where the index parameter k for the rectangle of size ρ_k is also the index parameter for the Markov process. A process state x_k at "time" k, will correspond geometrically to the end point of a boundary sequence passing out of R_k. Fig. 13 shows all possible locations for x_k, denoted as x_k^j, when $k=1,2,\cdots,5$. The number of possible states at time k is 1 for $k=1$, 3 for $k=2$, and $9+4(k-3)$ for $k\geq 3$.

Note that there is only one edge sequence between any two states x_{k-1}^i and x_k^j which is consistent with the generation model and which does not pass through another state x_k^l. As a result, a boundary segment $\{t_i\}_1^M$ is uniquely characterized by a state sequence $\{x_k\}_1^N$.

Fig. 14 contains an abstract representation of a typical realization of the Markov process, together with a description of the picture or character C_k associated with each state. The observed image will be a noise corrupted version of each such picture.

If the regions of interest have smooth, low curvature boundaries then a reasonable rule for assigning transition probabilities $p(x_k|x_{k-1})$ is to choose $p(x_k|x_{k-1})$ to be inversely related to the distance (measured in edge elements) between states x_{k-1} and x_k. We must also impose the total probability constraint that

$$\sum_{j=1}^{9+4(k-3)} p(x_k^j|x_{k-1})=1. \tag{62}$$

B. A Dynamic Programming Algorithm for Estimating Boundary Segments

Using the pixel data in an $N\times 2(N-1)$ block, R_N, we next formulate a dynamic programming algorithm for estimating the most likely state sequence consistent with the generation model. The algorithm is optimal in the sense that it finds the state sequence that maximizes the joint likelihood of the data in R_N and the corresponding edge sequence through R_N.

To begin we first define the pixel data sets

$$D_k=\{g_{ij}\colon \text{pixel }(i,j)\subset R_k\}$$

$$d_k=\{g_{ij}\colon \text{pixel }(i,j)\subset R_k,\ \text{pixel }(i,j)\not\subset R_{k-1}\}.$$

This implies that $D_k=D_{k-1}\cup d_k$, $D_1=$empty set. This recursion is essential. Next let $l(\cdot)$ denote a log-likelihood function, and $S_N=\{x_k\}_1^N$ denote a boundary state sequence of length N. Then $l(D_N,S_N)$, the joint log-likelihood of a boundary state sequence and the picture data, must satisfy

$$l(D_N,S_N)=l(D_N|S_N)+l(S_N) \tag{63}$$

where $l(S_N)$ is the log-likelihood of the state sequence S_N and $l(D_N|S_N)$ is the pixel data log-likelihood conditioned on the boundary $\{t_i\}_1^M$ described by S_N. Since the state sequence S_N is a Markov chain we can use

$$l(S_N)=l(S_{N-1})+\ln p(x_k|x_{k-1})$$

$$l(S_1)=l(x_1)=\ln P_s(x_1) \tag{64}$$

where $P_s(x_1)$ is the probability of a particular starting state x_1. Since boundary edge sequences are prohibited from reentering rectangles they have already passed out of, we can express

$$l(D_N|S_N)=l(D_{N-1}|S_{N-1})+l(d_N|x_N) \tag{65}$$

where $l(d_N|x_N)$ is the log-likelihood of the data added in extending the state sequence S_{N-1} to S_N conditioned on the specific new state x_N. Substitution of (65) and (64) into (63) leads to the following recursive expression for $l(D_N,S_N)$:

$$l(D_N,S_N)=l(D_{N-1},S_{N-1})+\ln p(x_N|x_{N-1})+l(d_N|x_N). \tag{66}$$

The transition probabilities $p(x_k|x_{k-1})$ can be calculated using a distance rule such as the one discussed above, while incremental data log-likelihoods, $l(d_k|x_k)$ can be calculated by observing that the pixel gray level values g_{ij} are $N(r_{\text{in}},\sigma^2)$ if g_{ij} lies inside the region and $N(r_{\text{out}},\sigma^2)$ when g_{ij} lies outside the region. Furthermore, once x_k has been specified, all pixel values g_{ij} in d_k can be associated with pixels either inside of or outside of the region. Hence if we define

$$f_G(x)\triangleq\frac{1}{\sqrt{2\pi\sigma^2}}\exp(-x^2/2\sigma^2) \tag{67}$$

we can use

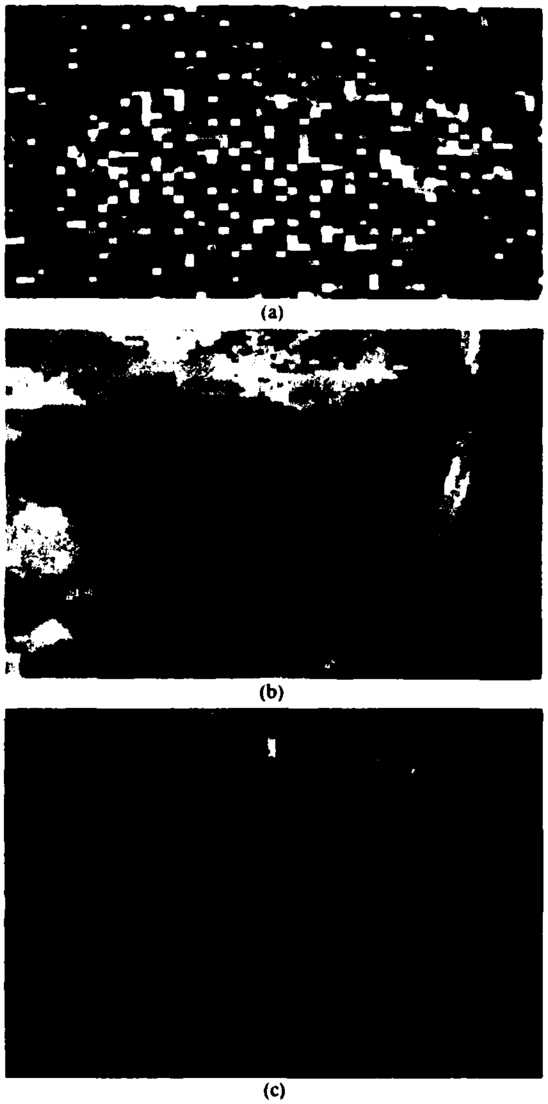


Fig. 15. Examples of algorithm performance on complete objects. (a) Ellipse with additive Gaussian noise such that $(r_{\text{in}}-r_{\text{out}})/\sigma=1$. (b) Lung section of human body, CAT scan. (c) Satellite image of a storm cloud.

$$\begin{aligned}l(d_k|x_k)&=\sum_{(i,j)\in I_{rk}}\ln f_G(g_{ij}-r_{\text{in}})\\&\quad+\sum_{(i,j)\in I_{bk}}\ln f_G(g_{ij}-r_{\text{out}})\\&=C-\sum_{(i,j)\in I_{rk}}\frac{(g_{ij}-r_{\text{in}})^2}{2\sigma^2}-\sum_{(i,j)\in I_{bk}}\frac{(g_{ij}-r_{\text{out}})^2}{2\sigma^2}\end{aligned} \tag{68}$$

where C is a constant which is independent of the choice of x_k, and

$$I_{rk}=\{(i,j)\colon \text{pixel }(i,j)\text{ is in region and }g_{ij}\in d_k\}$$

$$I_{bk}=\{(i,j)\colon \text{pixel }(i,j)\text{ is in background and }g_{ij}\in d_k\}.$$

Finally, a dynamic programming algorithm for estimating a state sequence S_N and hence a boundary edge sequence $\{t_i\}_1^M$ which maximizes $l(D_N,S_N)$ can be derived by observing that

$$\max l(D_N, S_N) = \max_{x_n} \left[\max_{S_{N-1}} l(D_{N-1}, S_{N-1}) + \ln p(x_N | x_{N-1}) + l(d_N | x_N) \right].$$

This boundary segment estimator has been incorporated into a complete boundary estimation scheme. At the last stage of the forward dynamic programming algorithm the three most likely states x_N are used to generate three complete paths out of a rectangle. These are stored as nodes on a tree, and an A^* types tree search algorithm [20] is used to piece together complete boundaries. Fig. 15 shows some examples of the overall algorithm performance. Fig. 15(a) shows the algorithm performance on an ellipse imbedded in Gaussian noise such that the signal-to-noise ratio $(r_{\text{in}} - r_{\text{out}})/\sigma = 1$. Both the actual and estimated boundaries are plotted. Fig. 15(b) shows the boundary obtained for a section of the CAT scan given in Fig. 12, and Fig. 15(c) shows the result of estimating the boundary of a satellite image of a cloud.

VII. Concluding Remarks

By constructing finite-state Markov chains, and assigning characters or observations to these states, one can model such things as continuous-phase FM signals and random boundaries in black and white images. The model may then be used to construct a likelihood function that may be written recursively and maximized with the techniques of dynamic programming. The resulting algorithms are tractable by nonlinear filtering and image processing standards, and the results often superior to what can be achieved with other approaches.

A great variety of signal and image processing problems may be phrased along the lines of this paper, and solved using the techniques of dynamic programming. This chain illustrates once again the great power of dynamic programming as a recursive optimization device.

Acknowledgment

The authors would like to thank M. Orfali and L. Srinivasan, graduate students at Colorado State University, for writing the software used to generate the examples given in Sections V and VI, respectively.

References

[1] H. Cox, "On the estimation of state variables and parameters for noisy dynamic systems," *IEEE Trans. Automat. Contr.*, vol. AC-9, pp. 5–12, Feb. 1964.

[2] H. Cox, "Estimation of state variables via dynamic programming," 1964 Joint Automat. Contr. Conf., Stanford, CA, June 1964.

[3] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 260–269, Apr. 1967.

[4] C. R. Cahn, "Phase tracking and demodulation with delay," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 50–58, Jan. 1974.

[5] G. D. Forney, Jr., "The Viterbi algorithm," *Proc. IEEE*, vol. 61, pp. 268–278, Mar. 1973.

[6] R. Bellman, *Dynamic Programming*. Princeton NJ: Princeton Univ. Press, 1957.

[7] R. E. Larson, *State Increment Dynamic Programming*. New York: American Elsevier, 1968.

[8] G. Goertzel, "An algorithm for the evaluation of finite trigonometric series," *Amer. Math. Monthly*, vol. 65, pp. 34–35, June 1958.

[9] R. Bellman, *Introduction to Matrix Analysis*. New York: McGraw-Hill, pp. 155–156, 1970.

[10] N. Levinson, "The Wiener RMS (root mean square) error criterion in filter design and prediction," Appendix B in N. Wiener, *Time Series*. Cambridge, MA: MIT Press, 1970.

[11] R. S. Bucy and A. J. Mallinckrodt, "An optimal phase demodulator," *Stochastics*, vol. 1, pp. 3–32, 1973.

[12] G. Ungerboeck, "New applications for the Viterbi algorithm: Carrier phase tracking in synchronous data-transmission systems," in *Proc. Nat. Telecomm. Conf.*, 1974.

[13] D. W. Tufts and J. T. Lewis, "Estimation and tracking of parameters of narrowband signals by iterative processing," *IEEE Trans. Inform. Theory*, vol. IT-23, no. 6, pp. 742–751, Nov. 1977.

[14] L. L. Scharf, D. D. Cox, and C. J. Masreliez, "Modulo-2π phase sequence estimation," *IEEE Trans. Inform. Theory*, vol. IT-26, pp. 615–620, Sept. 1980.

[15] L. L. Scharf and H. Elliott, "A random sampler of dynamic programming application in signal processing and control," in *Proc. 13th Asilomar Conf. Circuits, Syst., and Comput.*, pp. 7–13, Pacific Grove, CA, Nov. 1979.

[16] J. J. Wolcin, "Maximum a posteriori estimation of narrowband signal parameters," *J. Acoust. Soc. Amer.*, vol. 68, no. 1, pp. 174–178, July 1980.

[17] A. J. Rockmore, "Input parameter determination for target tracking," in *Proc. Advances in Acoust. Passive Target Tracking Conf.*, Monterey, CA, May 1977.

[18] M. H. Hueckel, "A local visual operator which recognizes edges and lines," *J. ACM*, vol. 20, Oct. 1973.

[19] H. Elliott, D. Cooper, and P. Symosek, "Implementation, interpretation and analysis of a suboptimal boundary finding algorithm," in *Proc. 1979 IEEE Comput. Soc. Pattern Recognition and Image Processing Conf.*, Chicago, IL, Aug. 1979.

[20] N. Nilsson, *Problem-Solving Methods in Artificial Intelligence*. New York: McGraw-Hill, 1971, pp. 54–70.

Louis L. Scharf (M'69–SM'76) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of Washington, Seattle, in 1964, 1966, and 1969, respectively.

Since 1971 he has been a member of the Electrical Engineering Department, Colorado State University, Fort Collins, where he is currently Professor of Electrical Engineering and Statistics. He enjoys drawing inferences from random data in real-time at high speeds. During the academic year 1974 he was a Visiting Associate Professor at Duke University, Durham, NC. He spent the academic year 1977 in France as a Professeur Associé at the University of Paris, South Orsay, and as a member of the Technical Staff in the CNRS Laboratoire des Signaux et Systèmes Gif-sur-Yvette. He has served as a Consultant to Honeywell, Inc., Seattle, The Applied Physics Laboratory, Seattle, and the Research Triangle Institute. He currently serves as an Associate Editor for the IEEE Transactions on Acoustics, Speech, and Signal Processing.

Dr. Scharf is a member of Eta Kappa Nu, Sigma Xi, and the Acoustical Society of America. He was Technical Program Chairman for the 1980 International Conference on Acoustics, Speech and Signal Processing.

Howard Elliott (S'75–M'78) was born in Boston, MA, on October 11, 1952. He received a combined Sc.B. and A.B. degree in electrical engineering and economics, the M.Sc. degree in electrical engineering in 1975, and the Ph.D. degree in electrical sciences in 1979, all from Brown University, Providence, RI.

He was appointed Post Doctoral Teaching and Research Fellow for one semester. He is presently an Assistant Professor of Electrical Engineering at Colorado State University, Fort Collins. His research interests are in multivariable and adaptive control, and digital image processing.

Dr. Elliott is a member of Tau Beta Pi, Sigma Xi, and Phi Beta Kappa.

APPENDIX B : Progress Reports and Miscellaneous Documents

PROGRESS REPORT

(TWENTY COPIES REQUIRED)

1. ARO PROPOSAL NUMBER: DRXRO-PR P-16437-EL

2. PERIOD COVERED BY REPORT: 1 September 1979 thru 31 December 1979

3. TITLE OF PROPOSAL: Viterbi Tracking of Randomly Phase Modulated Data

4. CONTRACT OR GRANT NUMBER: DAAG29-79-C-0176

5. NAME OF INSTITUTION: Colorado State University

6. AUTHOR(S) OF REPORT: Louis L. Scharf

7. LIST OF MANUSCRIPTS SUBMITTED OR PUBLISHED UNDER ARO SPONSORSHIP DURING THIS PERIOD, INCLUDING JOURNAL REFERENCES:

a. L. L. Scharf and H. Elliott, "A Random Sampler of Dynamic Programming Applications in Signal Processing and Control," 13th Annual Asilomar Conference on Circuits, Systems, and Computers, Nov. 5-7, 1979.

b. L. L. Scharf and H. Elliott, "Aspects of Dynamic Programming in Signal and Image Processing," IEEE Trans. on Autom. Control (submitted Dec. 1979).

c. O. Macchi and L. L. Scharf, "A Dynamic Programming Algorithm for Phase Estimation and Data Decoding on Random Phase Channels," IEEE Trans. on Inform. Theory (submitted Dec. 1979).

8. SCIENTIFIC PERSONNEL SUPPORTED BY THIS PROJECT AND DEGREES AWARDED DURING THIS REPORTING PERIOD:

a. Louis L. Scharf (not actually supported during this period)
b. Helen Anderson, M.S. student
c. Kazam Kazampur, M.S. student
d. Freddie Hanson, Work-study
e. David C. Farden, Ph.D. - consultant

16437-EL

LOUIS L. SCHARF
COLORADO STATE UNIVERSITY
ELECTRICAL ENGINEERING DEPARTMENT
FT. COLLINS, CO 80523

BRIEF OUTLINE OF RESEARCH FINDINGS

We are pursuing research on three distinct but related problems: (1) phase model extension to include random phase modulation, random FM modulation, and random chirp modulation; (2) frequency estimation in signal-plus-noise and autoregressive models; (3) dynamic programming algorithm development for FM tracking; and (4) simultaneous phase tracking and data decoding on random phase channels.

(1) Phase Model Extension: Here we have derived phase models for random phase, random FM, and random chirp modulation. Each model is a Markov chain defined on cyclic group. Covariance and spectral results have been derived. The results - not yet published - generalize existing results on the spectral theory of chains, and leave us with the problem of selecting states, transition probabilities, and "run lengths" to achieve model matching with more conventional models.

(2) Frequency Estimation: We have derived maximum likelihood frequency estimators and Cramer-Rao bounds for estimating frequency in complex normal signal-plus-noise and autoregressive models. The estimators have been simulated and modulo-2π errors studied. The results explode a currently popular myth regarding frequency tracking at low signal-to-noise ratios. Work will probably be published shortly.

(3) Dynamic Programming Algorithm Development: In reports (a) and (b) under item 7 of this document we have derived a dynamic programming algorithm for picking the optimum frequency track through a sequence of contiguous FFT maps to decode the MAP frequency sequence. Algorithm properties are under study and software development will begin soon.

(4) Simultaneous Phase Tracking and Data Decoding: A principle of optimality for phase tracking/data decoding has been derived and implemented in software to decode data symbols transmitted over random phase channels. Algorithm performance is treated in report (c) under item 7 of this report. The algorithm - though complex - outperforms all competitors.

PROGRESS REPORT

(TWENTY COPIES REQUIRED)

1. ARO PROPOSAL NUMBER: DRXRO-PR P-16437-EL

2. PERIOD COVERED BY REPORT: 1 September 1979 thru 30 June 1980

3. TITLE OF PROPOSAL: Viterbi Tracking of Randomly Phase Modulated Data

4. CONTRACT OR GRANT NUMBER: DAAG29-79-C-0176

5. NAME OF INSTITUTION: Colorado State University

6. AUTHOR(S) OF REPORT: Louis L. Scharf

7. LIST OF MANUSCRIPTS SUBMITTED OR PUBLISHED UNDER ARO SPONSORSHIP DURING THIS PERIOD, INCLUDING JOURNAL REFERENCES:

   a. L. L. Scharf and H. Elliott, "A Random Sampler of Dynamic Programming Applications in Signal Processing and Control," 13th Annual Asilomar Conference on Circuits, Systems, and Computers, Nov. 5-7, 1979.

   b. L. L. Scharf and H. Elliott, "Aspects of Dynamic Programming in Signal and Image Processing," IEEE Trans. on Autom. Control (submitted Dec. 1979).

(con't. on attachment)

8. SCIENTIFIC PERSONNEL SUPPORTED BY THIS PROJECT AND DEGREES AWARDED DURING THIS REPORTING PERIOD:

   a. Louis L. Scharf
   b. Claude Gueguen, Visiting Professor
   c. David C. Farden, Ph.D. - consultant
   d. Helen Anderson, M.S. awarded May 1980
   e. Freddie Hanson, Work-study

Dr. Louis L. Scharf 16437-EL
Colorado State University
Electrical Engineering Department
Fort Collins, CO 80523

7. (con't).

c. O. Macchi and L. L. Scharf, "A Dynamic Programming Algorithm for Phase Estimation and Data Decoding on Random Phase Channels," IEEE Trans. on Inform. Theory (submitted Dec. 1979; accepted April 1980).

d. L. L. Scharf, "Dynamic Programming for Phase and Frequency Tracking," NATO Advanced Study Institute on Underwater Acoustics and Signal Processing, Copenhagen, Denmark, 18-29 August 1980.

e. C. Gueguen and L. L. Scharf, "Exact Maximum Likelihood Identification of ARMA Models: A Signal Processing Perspective " Invited paper, EUSIPCO-80, First European Signal Processing Conference, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, September 16-19, 1980.

f. L. L. Scharf, D. D. Cox, C. J. Masreliez, "Modulo-2π Phase Sequence Estimation," IEEE Trans. Inform. Theory (to appear Sept. 1980).

PROGRESS REPORT

(TWENTY COPIES REQUIRED)

1. ARO PROPOSAL NUMBER: DRXRO -PR P- 16437 - EL

2. PERIOD COVERED BY REPORT: Through 31 December 1982

3. TITLE OF PROPOSAL: Viterbi Tracking of Randomly Phase Modulated Data

4. CONTRACT OR GRANT NUMBER: DAAG 29 - 79 - C - 0176

5. NAME OF INSTITUTION: Colorado State University

6. AUTHOR(S) OF REPORT: Louis Scharf

7. LIST OF MANUSCRIPTS SUBMITTED OR PUBLISHED UNDER ARO SPONSORSHIP DURING THIS PERIOD, INCLUDING JOURNAL REFERENCES:
   a. LL Scharf and H Elliott, "Aspects of Dynamic Programming in Signal Processing and Control, IEEE Trans on AC, 26, pp 1018-1029 (Oct 81)
   b. O. Macchi and LL Scharf, "A Dynamic Programming Algorithm for Phase Estimation and Data Decoding on Random Phase Channels," IEEE Trans on IT, 27, pp 581-595 (Sept 81)
   c. LL Scharf, CJ Gueguen, and JP Dugre, Parametric Spectrum Modelling: A Signal Processing Perspective," ASSP Workshop, Hamilton (Aug 81)

8. SCIENTIFIC PERSONNEL SUPPORTED BY THIS PROJECT AND DEGREES AWARDED DURING THIS REPORTING PERIOD:

   a. LL Scharf
   b. JP Dugre, Ph.D., July 1981

N.B. Copies of abstracts follow.

Dr. Louis L. Scharf 16437-EL
Colorado State University
Electrical Engineering Department
Fort Collins, CO 80523

BRIEF OUTLINE OF RESEARCH FINDINGS

The last progress report contained a complete list of accomplishments and ongoing work. That outline remains in force, with the addition of the following

a. Phase Model Extension. We are in the process of writing up our work on phase models on the circle. This work could lead the way to filtering on finite groups, a topic I raised to ARO in a letter to Suttle a year ago.

b. ARMA Systems. We have reformulated the autoregressive moving average (ARMA) modelling problem in terms of linear transformations, rather than linear filters. It's too early too give a prognosis, but new insights are developing. An invited paper for IEEE Trans on ASSP is in progress.

CSU

Department of Electrical Engineering

Colorado State University
Fort Collins, Colorado
80523

June 11, 1980

Dr. Jimmie R. Suttle, Director
Electronics Division
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709

Dear Dr. Suttle:

Here is my brief report on scientific accomplishments.

PROJECT: Viterbi Tracking of Randomly Phase Modulated Data
DAAG29-79-C-0176

OUTLINE: At Colorado State University the principal investigator and his associates are working on a nonlinear smoothing theory for randomly phase- and frequency-modulated information. The investigator's phase tracker has been generalized to a frequency tracker. Simulation and theoretical performance evaluations are in progress. Analytical investigation of Markov chains as approximants to FM signals is proceeding.

Application of these results arise in 1) detection and estimation of feeble sinusoidal signals (such as oscillation modes), 2) phase synchronization of data transmission systems, and 3) decoding of frequency-hopped FM signals.

INTERACTIONS: Irv Kullback, U.S. Army personnel, and USC Research group at Ft. Monmouth, May 29, 1980. (Meeting organized by Dr. Wm. Sander , ARO).

Louis L. Scharf
Professor
Electrical Engineering

LS:fr

Recent Outstanding Accomplishments : DAAG29-79-C-0176

July 12, 1982

The problem of FM demodulation has a long history of research and development in electrical engineering. In its modern form the problem is to estimate phase or frequency sequences from noisy data and to use these estimates in conventional and spread spectrum communication systems.

The principal investigator and his associates have developed models for random phase and frequency sequences and derived likelihood expressions for noisy observations of them. The investigators have applied dynamic programming to find an algorithm for computing the maximum of the likelihood. The algorithm has been applied to the decoding of binary, phase-shift-keyed, and quadrature-shift-keyed data sequences.

The results of this research suggest that there are numerous nonlinear filtering problems in signal and image processing that can be formulated and solved as nonlinear sequence estimation problems. Among the possibilities are boundary estimation in noisy black and white images, tomographic image reconstructive in noisy CAT Scans, and vehicle tracking from incomplete and noisy measurements.

APPENDIX C : Army Sponsored Meetings Attended by PI

The Principal Investigator attended the following ARO-sponsored meetings at Fort Monmouth:

Spread Spectrum, 29 May 1980
Fort Monmouth, New Jersey

Spread Spectrum Seminar, 22 May 1981
Fort Monmouth, New Jersey

At the 22 May 1981 meeting he presented a paper titled,

"Viterbi Tracking of Randomly Phase-Modulated Date"

TENTATIVE AGENDA

SPREAD SPECTRUM SEMINAR
Fort Monmouth, NJ

May 22, 1981

0830 Army Presentations

1030 Break

1045 Spread Spectrum Receiver Using SAW Devices
Prof. Pankaj Das, Rennselaer Polytechnic Insitute

1130 Viterbi Tracking of Randomly Phase Modulated Data
Prof. Louis Scharf, Colorado State University

1215 Lunch

1315 Research in Digital Communications
Prof. Robert Scholtz, University of Southern California
Prof. William Lindsey, University of Southern California

1445 Break

1500 Spread Spectrum Communications
Prof. Michael Pursley, University of Illinois
Prof. Robert McEliece, University of Illinois

1630 Closing